A PC BASED SOFTWARE FOR DETAILED VEHICLE MOVEMENT DATA ANALYSIS USING LABWINDOWS/CVI

A Thesis Submitted

in Partial Fulfillment of the Requirements

for the degree of

Master of Technology

by

FIRDAUS ANJUM KHALIL

to the

DEPARTMENT OF CIVIL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

July, 1996

CERTIFICATE

This is to certify that the work contained in this thesis entitled *"A PC based Software for Detailed Vehicle Movement Data Analysis using Labwindows/CVI"* , submitted by Fridaus Anjum Khalil in partial fulfillment of the requirements for the award of Master of Technology is a bonafide record of research carried under my supervision and guidance and that this work has not been submitted elsewhere for the award of a degree.

19 July, 1996

(Dr. S. P. Palaniswamy)
Professor
Department of Civil Engineering
Indian Institute of Technology
Kanpur - 208016

ACKNOWLEDGMENTS

I express my deepest sense of gratitude to my thesis advisor Dr. S. P. Palaniswamy for his constant help and encouragement in the successful completion of this work. I am grateful to him for making available the latest developments in software and hardware facilities, which have in turn led to my increased knowledge and interest in this field. I would like to thank Dr. Sanjay Gupta for providing me with details of the hardware design for this project. I am grateful to all my teachers at I.I.T Kanpur for helping me in shaping my professional skills in the field of Transportation Systems Engineering.

My sincere thanks to Mr.Bhuvanesh Singh and Mr.Hari Kishan Reddy for their valuable suggestions throughout my course work. I would also like to thank Ms. Alpana for her help in getting me acquainted with LabWindows/CVI.

I thank all the people who have been friends and helped me during my stay at I.I.T. Kanpur. In particular, I thank my lab mate Krishna Mohan who has been a constant companion all through my work and has made my stay lively and cheerful. His company during the long hours and the late nights in the lab was memorable. I would like to thank my hostel mates Ravinder Reddy, Rajesh, Kali, Ravi, Sandeep, Murthy and Avadhesh. I would never forget the memories of all the interesting, endless chats we shared during our long evening strolls.

I can never thank enough my parents and sisters. Their inspiration, expectation and confidence have been a pillar of strength to me all the time.

Firdaus Anjum Khalil

I wish to thank Dr. Sanjay Gupta, Chief Scientific Officer, Advanced Center for Material Sciences, for his involvement and expertise which has been vital for this project.

S. P. Palaniswamy
(Thesis Supervisor)

ABSTRACT

Although road transport plays an important role in the well being of our country, it is the most neglected sector of the economy. The deteriorating road conditions and lack of proper infrastructure have resulted in rapid increase in accidents. Need has been felt by the transport operators for a scientific and planned way of fleet management and maintenance. The operations of trucks are suboptimal and there is no control on its movement once it leaves the godown. Thus a system is required which could help in monitoring and controlling the vehicles on roads.

The present work is an attempt to integrate the developments in instrumentation and computer technology to address the road traffic monitoring and surveillance problems.

The objective of the work presented in this thesis is to develop a software for the analysis of detailed vehicle movement data. LabWindows/CVI has been used as the software tool to develop the Graphical User Interface for this software. The analysis of the data helps in predicting the behavior of the driver to different situations on the road.

The software has been developed to analyze data collected from the data recorder fitted to the vehicle. The data is transmitted by the sensors fitted to the speedometer cable inside the gear box and stored in the microprocessor after filtering and necessary modifications. The microprocessor is located in the passenger cabin.

The software has been developed in two stages. In the first stage, the program logic to analyze the various attributes of the travel was developed. This was developed using C programming language. In the second stage Graphical User Interface was added to the program to make it user friendly.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Like the 'Flight Data Recorder' mounted on aircraft to record flight data such as speed, acceleration, altitude, etc., need has been felt to have a similar monitoring system on a surface transport mode like trucks and buses, to record distance traveled, time of travel, fuel consumed, etc. Such devices known as "Tachographs" are mandatory in the developed world. While almost all Tachographs are electro mechanical, with the corresponding problems, it is therefore attempted to develop a fully electronic(microprocessor controlled) system which is likely to be cheaper, more robust and easy to use.

With the increasing accident rates, the need for better surveillance and monitoring has increased. About 60 thousand persons are killed in road accidents every year in India. Overspeeding and ignorance of road rules compounded with poor road conditions are the major factors for this high accident rate. Therefore it is necessary to evolve suitable measures to minimize the loss of lives on our roads.

The analysis of detailed vehicle movement data could provide information about the driver's behavior throughout the journey. The operators can also monitor the condition of their vehicles and components from time to time. The analysis of accidents could help the insurance agency to settle insurance related disputes and also to determine the authentication of the claim made. At present there are no such existing system.

The present work attempts to integrate the latest developments in the field of instrumentation and computer technology to address the road traffic speed monitoring and

enforcement problem. This work is an attempt to initiate a process of evolving a software for complete analysis of the vehicle movement data.

1.2 Current Scenario

Road transport plays a pivotal role in Indian Economy. Trade and commerce are totally dependent on it. Major share of goods transport is accomplished by road transport. Other means of transport such as rail and water transport also rely on road transport. The health of the road transport has a far reaching effect on the whole system.

However, inspite of its visible importance, road transport has not seen much improvement in its conditions. With the liberalization of economy, better equipped vehicles are replacing vehicles which were in use for such a long time. In case of heavy surface transportation, the concept of containerization is gaining momentum in our country. These changes compounded with a need for a more organized and scientific way of working have resulted in need for a better monitoring system, especially by transport operators.

1.3 Need for Monitoring and Surveillance System

Unlike Flight Recorders fitted in airplanes, which records data such as speed, acceleration, headings etc. which are universally used, there are no such system in widespread use in surface transport in the developing countries, wherein the travel characteristics of vehicles could be recorded and analyzed in the wake of an accident or for purely monitoring and surveillance purposes. Imported units were expensive (Rs. 75,000 and above) and not rugged enough to withstand the harsh work environment.

If such a system were available at moderate cost, the transport operators who maintain a huge cache of vehicles, could keep track of each one of their vehicles at any instance of time. This would certainly help in improving the efficiency of the vehicle fleet and more importantly it would promote an efficient transport network, especially in goods transport.

If an accident occurs, data such as speed characteristics, acceleration variations, fuel efficiency, temperature of various components etc., give an insight into its possible causes. For example, the normal sign of wheel failure is heating up of the axle. Hence, this could serve as a warning to the driver during the journey and also to diagnose on the possible causes of accidents. Accidents could be due to mechanical failure or the drivers negligence. In both the cases, availability of relevant data could be very useful in its diagnosis. This could also be used by the operators to identify and educate the erring drivers. This in turn could help to reduce the rapidly rising rate of accidents in our country.

The macro flow chart depicting the existing problems and the need for a vehicle monitoring and surveillance system has been shown in Figure 1.1. the transport problem in our country could be broadly classified as Infrastructure related, traffic related and Environmental related. In our country most of the people related to the of transportation sector are illiterate. They do not undergo any sort of training and have little or no knowledge about the traffic rules. This along with the other reasons listed in Figure 1.1, results in higher road accidents.

Thus a need for a suitable vehicle monitoring and surveillance system which incorporates a scientific approach has been felt by the transport industry. The present work is a small step towards addressing the traffic related problems and thus helping in initiating a scientific approach to solving the transport problems.

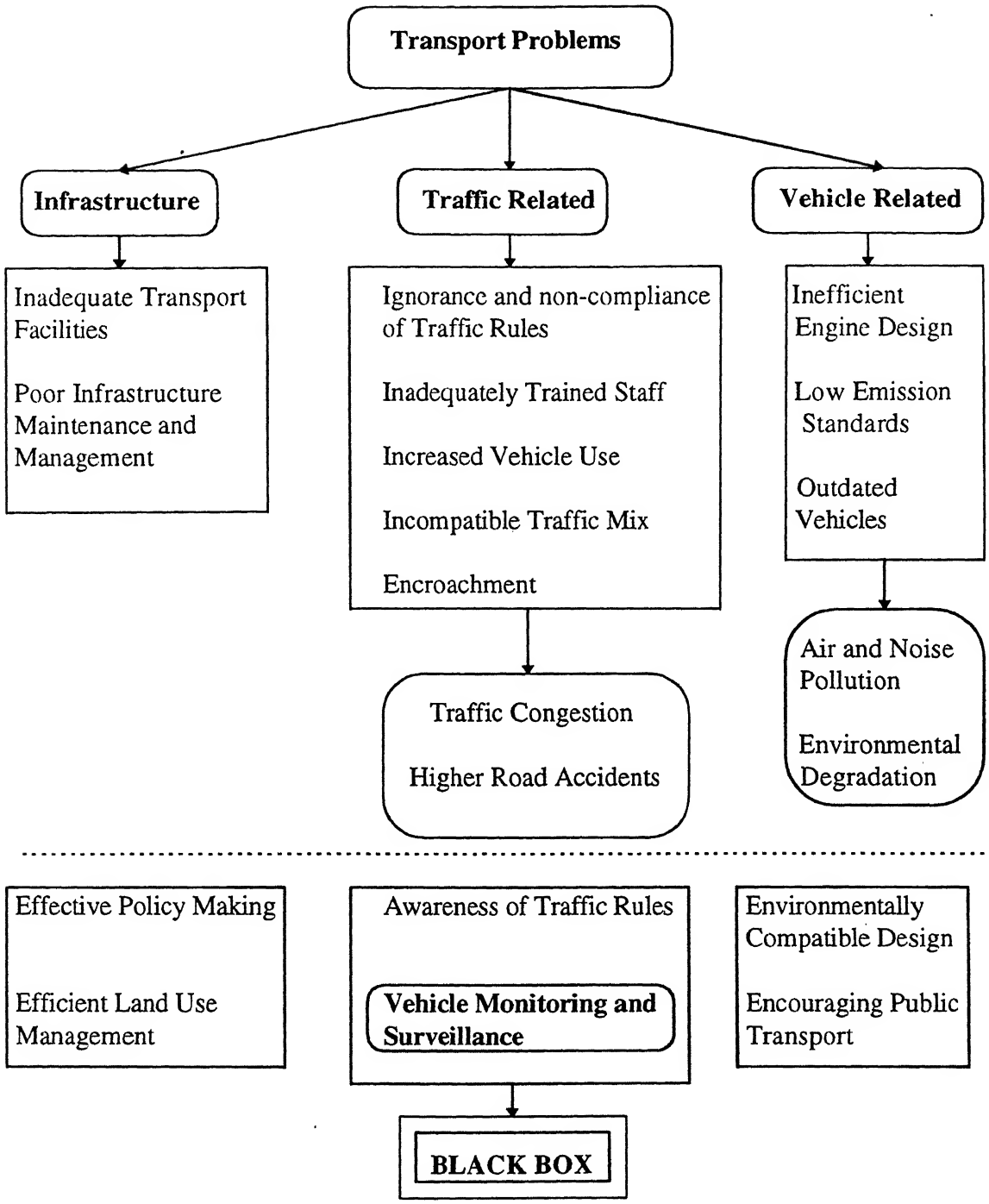

Fig. 1.1 Flow Chart of the Transport Problems and Utility of Black Box

1.4 Problem Definition

The present work enumerates a logical flow of data throughout the program. It gives an approach as to the various steps that can be taken to analyze the data obtained from the vehicles.

Data is transferred from the 'microprocessor' through RS-232 ports by the computer as streams of pulses of a definite time interval, under interrupt conditions and stored in a file. Sets of data are read from the file and analyzed. The final analysis is in terms of speed characteristics, acceleration variations, fuel consumption at various stages, temperatures of various components etc. the whole software has been written in C programming language('LabWindows/CVI') in the Virtual Instrument environment. A built in function of the software tool to fit Polynomial Equation to curves has been made use of. The program is divided into several modules. LabWindows/CVI is an ANSI-C compiler with very powerful GUI support.

The analyzed data could give information about the reaction of the driver during the vital seconds just before an accident. This could further be used to suggest the probable causes of the accident.

1.5 LabWindows/CVI -- Aid in User Interface Building

LabWindows/CVI is a software development system for C programmers. It contains an interactive environment for developing programs as well as libraries of functions for creating data acquisition and instrument control applications. LabWindows/CVI contains a comprehensive set of software tools for data acquisition, analysis and presentation.

The interactive environment is made use for editing, compiling, linking and debugging ANSI C programs. Also the functions in the LabWindows/CVI function libraries can be used extensively to write programs. In addition each function has an interface called a *function panel* complete with on-line help that guides the user to interactively execute the function and generate code for calling the function. Programs written within the

LabWindows/CVI interactive environment must adhere to the ANSI C specification. In addition, the user is free to use compiled C object modules, Dynamic Link Libraries (DLLs), C libraries, and instrument drivers in conjunction with ANSI C source files when developing a program.

LabWindows/CVI enhances the C programming language for instrumentation applications. In the present work, this software tool has been used extensively to build a Graphical User Interface. However to use this software, a thorough knowledge of C programming language is necessary.

1.6 Organization of the Thesis

In this introductory chapter the objective behind this thesis is elaborated. A brief outlay of how the problem is tackled along with an overview of the software tool LabWindows/CVI is also summarized. A flow chart depicting the transport problems and need for the present work is also described. The second Chapter deals with the literature reviews about Flight Recorders and interview responses from the transport operators and insurance agency. The third chapter discusses LabWindows/CVI in detail with regard to its utility, features and limitations. Examples of its use in various projects have also been cited. In the fourth chapter a discussion on data acquisition and hardware design are explained in detail. The types of sensors to be used and their characteristics have been elaborated and the procedure for down loading data into the computer is discussed. Chapter five gives the flow chart to show the flow of data into various modules of the software, alongwith details of each of the modules. Block diagrams of each of these modules are also shown. The front panel of the software is discussed in detail giving the utility of each of the control buttons in the panel. Chapter six details the discussions of the result of the example data analyzed, conclusions and scope for future work.

Chapter 2

LITERATURE REVIEW

2.1 Interview Responses

The basic aim of this software is to provide a scientific method of data acquisition and analysis to enable the transport operators to monitor their vehicles in an effective manner. Keeping this in mind the software has thus been developed from the point of view of its utility to the operators. In this regard, transport operators were interviewed. The President of the Kanpur Transport Operators Union, Mr. Shyam Gupta was interviewed to get ideas for the orientation of the software. Dr. Sanjay Gupta, Chief Scientific Officer, IIT Kanpur, talked to Mr. N.C.Mittal, C. Engg. connected with the Insurance industry, to know about an insurance man's point of view as to how this software could help in speeding up the insurance process in the event of an accident. The following enumerate the important features of the responses from the operators. Attempt has been made to incorporate most of the features listed below.

- **Responses from Mr. Shyam Gupta**

The most basic concern of the transport operators were the accidents and reliability of their drivers. The operators were interested in four important aspects that would help them in managing their vehicle fleet better. They are as follows:

* total distance traveled by the vehicle
* total fuel consumed by it to undertake this journey and also the amount of oil used
* load carried by the vehicle at different times during its journey and most importantly
* the cause of accidents, if any.

Most of these aspects were related to the reliability of the drivers. Since the built in odometer could be tampered with, they couldn't rely on the drivers to give exact kilometerage.

More so, since fuel consumed depends upon the distance traveled, the operators concern was to determine the exact amount of fuel consumed. The fuel consumption data also helps to gauge the performance of the vehicle throughout the journey. It also gives insight into the pilferage of fuel.

Once the vehicles leave the warehouse the operator is unable to follow the vehicle, during its journey. The vehicle carries a certain amount of load when it starts off from the warehouse, however the drivers in their quest to earn money put additional load on the vehicles along the way. This leads to overloading and may result in accidents. Due to overloading the lifespan of the vehicle and its parts also decreases. This in a way is a loss to the operators.

Accidents were a major cause of worry to the transport operators. According to the operators, if the possible causes of accidents could be ascertained, it would help in identifying whether the accidents were caused by the mechanical failure or due to the negligence of the driver. This would help them keep track of the erring drivers. Apart from the above mentioned concerns voiced by the operators, the final cost of the system was also a major concern for the operator.

- **Responses from Mr. N C Mittal**

According to Mr. Mittal, if the system is not too expensive, the insurance industry would like to make it mandatory as it would help in settling the disputes regarding insurance claims. It would also help in verifying the authenticity of the claims.

2.2 Flight Recorders - Source of Inspiration

The inspiration for the present work has been drawn from the flight recorder commonly known as the 'Black Box'. Like the Black Box, this represents a mechanism that records relevant signals from different parts of the vehicle during operation. The signals, of a definite time interval transmitted from these parts, are stored in the recorder. The duration of the data stored in one run depends upon the memory of the mechanism. The present work is based on the understanding that the data for the entire duration of the journey is available for analysis. To understand the working of such a mechanism a brief summary of the flight recorder is given below.

2.3 The Black Box

Flight recorders or 'Black Box'[1] as they are popularly known, are instruments that are fitted on aircraft. They record signals from various transducers sensing air speed, altitude, engine speeds, temperature etc., in a repetitive sequence, usually once per second, onto a recording medium. The first flight recorders were introduced in the late 1950s, on military aircraft as a result of not knowing the precise cause of air crashes. Contrary to popular belief, the flight recorders are not black boxes in the true sense. They are painted bright orange with white reflective strips to assist locating them at crash site.

2.3.1 Flight Recorders Development and Latest Technology

Flight recorders first introduced used stainless steel spools as the recording medium. Later on thick metal foils on which scribe were attached to moving coil meters and air pressure mechanisms, literally scratched traces into the moving foil medium. The foil moved at a rate of six inches per hour, thus providing an accurate time reference.

As recorders were developed, magnetic tapes became the most popular medium. As the field of electronics developed, an additional avionics box known as Flight Data Acquisition Unit was introduced. The new box collected parameters from a large variety of sensors and systems on the aircraft. These were sorted and labeled into digital format, transmitted to flight recorder over a 2-wire serial bus [1,2,3].

The intricate tape mechanisms being time consuming and expensive to maintain are being replaced by large capacity computer memory chips and are termed 'solid-state' because they have no moving parts. This increases their reliability, require minimal maintenance and can survive harsher fire and impact conditions. Both 'solid-state' Flight Data Recorder and Cockpit Voice Recorder are available, and consists of banks of memory chips mounted in a crash protected enclosure; decoding and controlling electronics; a power supply and an ultra sonic locator beacon.

2.3.2 Types of Flight Recorders

At present, passenger aircraft carrying over a certain weight categories are required to carry two different Flight Recorders. These are Cockpit Voice Recorder (CVR) [1] and the Flight Data Recorder (FDR) [1,2]. Helicopters and military aircraft are permitted to use a Combined Voice and Flight Data Recorder (CVFDR) [1].

Cockpit Voice Recorder - The CVR is essentially a crash-protected 4-track voice recorder which records the audio from the pilot and co-pilot, as well as passenger announcements and the signal from the area-microphone mounted on the cockpit ceiling. Typically, each of these channels is stored on a 30 minute recording loop. After this the audio is overwritten so that the CVR will always store the last half-hours speech.

Flight Data Recorder - The FDR records performance data relating to the airframe, engines and avionics systems. These parameters differ with aircraft type but will generally contain information such as time, vertical acceleration, altitude (pitch and rolling altitude),

thrust (each engine), speedbrake position, autopilot engagement, longitudinal and lateral acceleration, ground speed, outside air temperature etc. These parameters are usually recorded for 25 hours, after which the recorder will recycle and overwrite the old data with new.

Combined Voice and Flight Data Recorder (CVFDR) [1] - On some aircraft, particularly helicopters and military fighter aircraft, it is an advantage to combine both the CVR and the FDR into a single box. Made possible by integrated electronic components and latest manufacturing techniques, this method has the benefit of saving weight and apace over the traditional two box system, yet records the same capacity of each.

2.3.3 Composition

The flight recorders generally consist of a titanium box lined with heat insulation within which the tape mechanism or solid-state memory module is housed. The heat insulation protects the recording medium from temperatures of 1100 °C. Materials used are sometimes polymer based, wax or gelled water, all of which can act as a heat barrier from the high temperatures. In addition to fire survivability, the recorder must be able to withstand shock, impact, crush, vibration, adverse pressures etc. The flight recorder can withstand an impact equivalent to reaching a dead stop from 360mph in only 16 inches.

2.3.4 Location

The flight recorders are typically located above the ceiling at the rear of the passenger cabin, underneath the tail fairing.

2.3.5 Monitoring Flight Safety

The Flight Data Recorder contains considerable information to determine the cause of an air accident. However, it is more useful to identify problem areas before they become

hazardous. To facilitate this, an additional non-crash protected recorder, namely the Quick Access Recorder (QAR) [1] is fitted into the aircraft. QARs store many more parameters, at higher sample rates and for much longer duration than the FDR. The data is replayed and analyzed on a ground based computer such that any faults or abnormalities can be discovered. The FDR can tell "what went wrong" whereas the QAR advises "what's going wrong".

Modern computers can perform detailed analyses of engine, autopilot, airframe and navigation systems etc., and problems and causes identified using the data provided by the flight recorders.

Although the utility of the instrument, for which this software has been developed, is similar to the 'Black Box', it is a simpler version of the same. It also receives pulses from the various sensors and the microprocessor stores it for a certain time period at a stretch, depending on its memory.

Chapter 3

INTRODUCTION TO LABWINDOWS/CVI

LabWindows/CVI [2,3,4,5] is a rich and powerful development environment, featuring libraries to aid in creating programs for a multitude of data acquisition, test and measurement applications. It is a comprehensive software development system for C programmers. It contains an interactive environment for developing programs as well as libraries of functions for creating data acquisition and instrument control applications. The built in libraries of LabWindows/CVI are very powerful, having functions for developing all phases data acquisition and instrument control systems. In addition the complete standard ANSI C library is available within the LabWindows/CVI development environment.

Some of the salient features of LabWindows/CVI development tool has been summarized below.

3.1 Features

* Interactive development environment for building instrumentation applications using the ANSI C programming language.
* Visual development tools automatically generate program outlines and function calls to minimize coding errors and accelerate program development.
* Integrated C programming tools including 32-bit C compiler, linker, debugger, and code-generation utilities.
* Intuitive graphical editor for building custom GUI displays
* Extensive analysis library for signal processing, statistics, curve-fitting, and complex analysis

* Networking capabilities with DDE and TCP/IP Libraries

With LabWindows/CVI (**C for Virtual Instrumentation)** virtual instruments can be built to match the specific needs of the applications. The integrated visual development environment makes it easy to combine data acquisition and instrument control hardware and software components into a PC-based virtual instrument. Using the open LabWindows/CVI development environment, it is easier to build virtual instruments with any instrumentation hardware available using C.

3.2 Visual Programming for Instrumentation

With LabWindows/CVI, the key to improved programming productivity is the visual development interface for building programs. With CodeBuilder, an automatic program-generation utility, C source code can be generated quickly, eliminating the hours of manual coding and debugging efforts required with traditional programming systems. CodeBuilder generates standard ANSI C source code that is easy to edit or enhance before compiling. The visual development tools in LabWindows/CVI, significantly reduce system time-to-market without sacrificing the power, flexibility, or performance of compiled C.

The instrumentation libraries and utilities built into LabWindows/CVI reduce many of the time-consuming programming tasks associated with traditional instrumentation system development. With the drag-and-drop User Interface Editor, GUI design is easy - no complicated graphics programming is required. Instrument drivers replace low-level instrument command programming and syntax with intuitive, high-level functions. Built-in hardware control libraries ensure trouble free integration with your GPIB, and plug-in DAQ instrumentation hardware.

3.3 Interactive C - A New Approach

LabWindows/CVI represents a new approach to instrumentation programming - an interactive approach. With LabWindows/CVI, the advantage of the power and performance of compiled ANSI C code can be utilized with an interactive, easy-to-use programming interface. LabWindows/CVI is an ideal environment for instrumentation systems. With function panels in LabWindows/CVI, C programmers can execute code line by line, and experiment with instrumentation hardware.

The interactive programming tools in LabWindows/CVI are fully integrated with traditional power C programming utilities. LabWindows/CVI has a built-in, 32-bit ANSI C compiler, so the programs run fast. In addition, LabWindows/CVI has a full-functioning source editor, linker, debugger, and variable display capabilities.

3.4 LabWindows/CVI Environment

The LabWindows/CVI Environment makes it easy to create and test applications that use the LabWindows/CVI libraries. The environment is a combination editor, compiler, and debugger with extensive run-time checking. A special feature, called a function panel, makes the task of developing programs much easier. Using a function panel, executing a LabWindows/CVI library function and generating codes can be done interactively. Function panels also contain on-line help information for the functions and function parameters. Building, execution, testing, and debugging the source code can be easily accomplished in the LabWindows/CVI environment.

The process of creating an application is made very easy and systematic by using the various interactive windows provided in the software. Some of these windows that have been used to develop this program are:

- Source windows:

Used to open, create, edit, run debug, and save source code. Source windows display the source code for the program that is being developed. The windows behave like standard text editors. The program can be typed in text form directly into a source window or load text from an ASCII file. Codes from LabWindows/CVI function panels can be inserted directly into source windows. A program can be saved from a source window as an ASCII file.

- Interactive Execution Window (IEW):

Used to execute selected portions of code. A complete program is not necessary in the IEW, as is the case in a source window. For instance, variable declarations and assignment statements in C can be executed without declaring a main function. The IEW is used to test a portion of code before including them in the main program.

- Variable Display, Array Display, String Display, and Watch Windows:

Used for debugging programs. These windows are made use of to inspect and modify the values of program variables. These windows may be evoked either when no program is running or when a program is suspended at a breakpoint. The names and types of all variables, including arrays and strings, are shown in the Variable Display windows. The current values of numeric scalars, values and contents of pointers, and strings contents are also shown in the Variable Display windows.

- User Interface Windows:

Used to build graphics-mode command bars, pull-down menus, dialog boxes, controls, graphs, and strip charts and save them to User Interface Resource (**.uir**) files. The User Interface Editor is described further in this Chapter in detail.

- Standard Input/Output Window:

Used for printing text messages and receiving user input from the keyboard.

3.5 User Interface Editor - An Overview.

The User Interface Editor[3] is an interactive graphical environment for designing custom GUIs for programs. It consists of a wide variety of controls, such as pushbuttons, knobs, meters, dials, gauges, text boxes, pull-down menus, graphs, and strip charts to enhance the presentation on the GUI. In addition, graphic images can be imported, such as PCX or BMP, onto the GUI to customize it for the system.

Each control type has a variety of attributes that can be set in intuitive dialog boxes by double-clicking on a control. By setting the attributes, full advantage of the built-in functionality of each control can be utilized for displaying and manipulating technical data. For example, the Edit Graph Control dialog box can be used to set graph attributes such as axes (auto scaled or logrithmic), plot style (line style, colour, marker style etc.) etc. It can also be used to set graph cursor attributes to interactive mode (as has been done in the present work) to directly read points from the graph using the cursors. In addition to setting the operation of the controls and indicators, their appearance, fonts, ranges, data type, and precision can be controlled in the User Interface Editor. With the LabWindows/CVI User Interface Editor, a great deal of user interface "programming" can be accomplished without writing a single line of code.

Editing and modification of a Graphical User Interface is done in the User Interface Editor. This editor is activated as a window panel on which the required GUI is built. This window contains a tool bar for high level editing with mouse. When a particular tool is clicked, the mouse cursor changes to reflect the new editing mode.

The following tool options are available:

- Editing Tool - to select position and size of the objects that are to be displayed on the GUI.

- Labeling Tool - to modify text associated with these objects.
- Coloring Tool - to color the objects created. A color palette is displayed which makes selecting the appropriate colors easier and also the changes made can be instantaneoûsly previewed.
- Operating Tool - to operate on the objects. While in this mode the control events are also displayed. The control events are explained later in the section.

3.5.1 Automatic Program Generation

Once a GUI is designed in the User Interface Editor, the CodeBuilder can be used to automatically generate program source codes. The result is a custom C program with callback functions set up to respond to user mouse-clicks and keypress events for each of the controls on the GUI.

When a GUI is designed, it essentially means defining areas on the screen in the form of controls that can generate events. A single mouse click on a command button will pass the following user interface to the program for processing:

1. GOT_FOCUS event - If the command button is not the active control, a click on this button will make it active control. When a control gets the input focus, a GOT_FOCUS event occurs.
2. LEFT_CLICK event - Clicking on the left mouse button results in this event control.
3. COMMIT event - After clicking and releasing the mouse button, a COMMIT event occurs signifying that the control button has been activated and committed.

Commit events are generated when the GUI actually commits to an operation such as pressing **<Enter>** from the keyboard. When a control is created, a commit event has to be assigned to it. The following control modes determines how the control generates commit events:

- Normal - specifies that the control may be used and can also be changed programatically.

- Indicator - specifies that the control may be changed programatically but a commit or value changed events cannot be generated.
- Hot - is identical to normal except that the control generates a commit event when acted upon it. Most of the command buttons used in the program makes use of this mode.
- Validate - is same as hot except that, before a commit event is generated, the program validates all numeric controls on the panel.

All the controls depicted in the front panel have one of the above listed control modes. The front panel in the present work utilizes the Hot and Indicator controls only. Once the User Interface Panel (**.uir**) is made, it automatically generates its corresponding header (.h) files.

The front panel is characterized by a panel name. All the other control units such as command buttons, menu bars, numeric controls, graphs, strip charts etc., are identified by the name of the panel followed by their constant names. For example if the panel is named as 'PANEL' and the strip chart as 'CHART', then in the program, the function which controls the strip chart is identified as 'PANEL_CHART'. Similarly a parent panel can have more than one subpanel. However, they are identified by the parent panel only.

To display a front panel on the screen programatically, a handle has to be assigned which loads the specified user interface generated in the editor. By using the RunUserInterface function (which is built in) the front panels can be displayed along with the various control units on it.

3.6 Application with Function Panels

The built-in LabWindows/CVI libraries have interactive code-generation tools that help to speedily generate function panels. With function panels, the LabWindows/CVI library functions can be experimented interactively, before using them in a program, to gain a

good understanding of how each function works. Finally, function panels automatically generate code and place it into the source file, jump starting coding efforts.

A function panel is a graphical representation of a LabWindows/CVI library function and its parameters. By entering values in the function panel controls, the function call syntax can be built interactively. A function panel can be used to insert acquisition and analysis code into the program shell to complete the application.

Due to the flexibility of generating the source code automatically in LabWindows/CVI, more reliable, standardized source code that has fewer programming problems and syntax errors to debug can be generated.

3.7 Instrument Drivers

Instrument drivers[5] are a key development tool in LabWindows/CVI, and are extremely useful for instrument control applications. An instrument driver is simply a set of high-level functions for acquiring data from and controlling GPIB, VXI, or serial instruments. With an instrument driver, programming with functions like Initialize, Configure, Measure, Read Waveform, and Close become easier instead of using cryptic ASCII command strings. In LabWindows/CVI, instrument drivers are accessed through function panels. As in any other LabWindows/CVI function panel, the control parameters are entered in the panel and the controls and the function call is built automatically at the bottom of the screen. The Read Waveform function builds an ASCII command string for transferring a waveform from the oscilloscope, sends the string over the GPIB, VXI, or serial interface, receives the waveform data from the oscilloscope, parses the data, scales the data, and places the data in an array in memory. The instrument driver acts as a high-level translator to simplify instrument programming tasks.

3.8 The Analysis Libraries

The LabWindows Analysis Libraries [5] offer a powerful, comprehensive set of more than 150 analysis functions for data acquisition and instrument control. The analysis functions give complete flexibility to develop applications in areas ranging from statistical process control to DSP. The LabWindows/CVI environment also has the advanced and the basic library utilities with the following features.

The Base Analysis Library (included with the LabWindows/CVI Base Package) gives a complete set of array manipulation, complex arithmetic, and basic statistical functions. The standard Analysis Library includes such functions as:

* 1D and 2D array addition, subtraction, multiplication, and division
* Linear evaluation, max/min, and array extraction
* Complex array addition, subtraction, multiplication, and division
* Polar-to-rectangular conversion
* Mean value, standard deviation, histogram, and sort
* Dot and cross product, matrix inversion, transpose, and determinant

In addition the Advanced Analysis Library, included with the LabWindows/CVI, delivers technology in high-performance analytical and computational algorithms. These algorithms are packaged in standard C function calls, providing maximum flexibility for integration into the built instrumentation applications. Each analysis function has a corresponding LabWindows/CVI function panel, which can be used to experiment with each of these functions interactively and view the results before incorporating the function call into the program. Built around the analysis functions are the measurement-based functions - a layer of high-level routines that simplify the integration of advanced signal processing algorithms into the instrumentation applications. These measurement based functions maintain engineering units and signal information throughout the computation process.

Some of the features of the Advanced Analysis Library are listed below.

* Fast Fourier Transform (FFT) and Fast Hartley Transform (FHT)
* Integration, differentiation, convolution, and correlation
* Power spectrum and pulse parameters
* Digital filters .
    - Finite Impulse Response-windowing and Parks-McClellan equi-ripple
    - Infinite Impulse Response -Butterworth, Chebyshev, and elliptic
* Windowing functions - Hamming, Hanning, Blackman, triangle, and Kaiser
* Signal generation
    - Pulse, ramp, sine, impulse, and triangle waveforms
* Uniform, white, and Gaussian noise
* Linear, exponential, and polynomial curve fit
* Variance, rms, moments, and median
* Complex log, exponential, power, and square root
* Advanced matrix operations

3.8.1 The Acquisition and Control Libraries

The real power of LabWindows/CVI is found in the hundreds of functions that make up the instrumentation libraries. Integrated among the language tools and development utilities in LabWindows/CVI are instrumentation libraries that simplify communication with GPIB (General Purpose Interface Bus), serial, and plug-in DAQ(Data Acquisition) hardware.

3.8.2 The RS-232 Library

The RS-232 Library has functions for performing serial communication using multiple RS-232 ports under interrupt control. The RS-232 is used to connect a computer to the external instruments and download the signals.

3.8.3 The Networking Libraries

With the LabWindows/CVI DDE and TCP Libraries, data or control processes can be passed across a network. A client or server applications can also be built in LabWindows/CVI.

3.9 Examples [2]

LabWindows/CVI is a new concept in C based virtual instrumentation which is being used extensively worldwide. More and more projects are being developed in most of the engineering and instrumentation fields. This software development tool has been used in areas such as Telecommunications, Automotive, Process Control, Aerospace, and Education. The University of Texas A&M are currently offering a one semester course only on LabWindows/CVI. A few examples of its use have been summarized below.

A portable vehicle monitoring system for the rail transit industry has been developed using LabWindows/LabView as the base. Various performance related characteristics are monitored on board the transit system. About 100 virtual instruments were made using this software which acquired signals from the various components of the transit system and analyzed.

Chapter 4

HARDWARE DESIGN

4.1 Hardware Design and Working

The hardware design and its working have been described briefly in the following sections. ITI Rae Bareilly are required to design the actual hardware to be used in the project.

The whole design is divided into three broad categories. They are listed below:

- Hardware for data acquisition
- Hardware for data storage and
- Software for data analysis.

The data acquisition is accomplished by the sensors or pickups installed at strategic location whereas the data storage is taken care of by the microprocessor unit. The analysis of this data is carried out by the software, in the PC.

A diagrammatic representation of the total setup of data flow from the sensors to the PC is presented in the flow chart shown in Figure 4.1.

The main data required in this project is the time and distance. The most logical way of acquiring the data set is from the region just before the main drive. This could be accomplished by connecting the sensors to the speedometer cable inside the gear box. Knowing the type and wheel size, final drive ratio and speedometer gear ratio, it is fairly straightforward to calculate the frequency of the pulses transmitted. The sample calculations have been shown later in this Chapter.

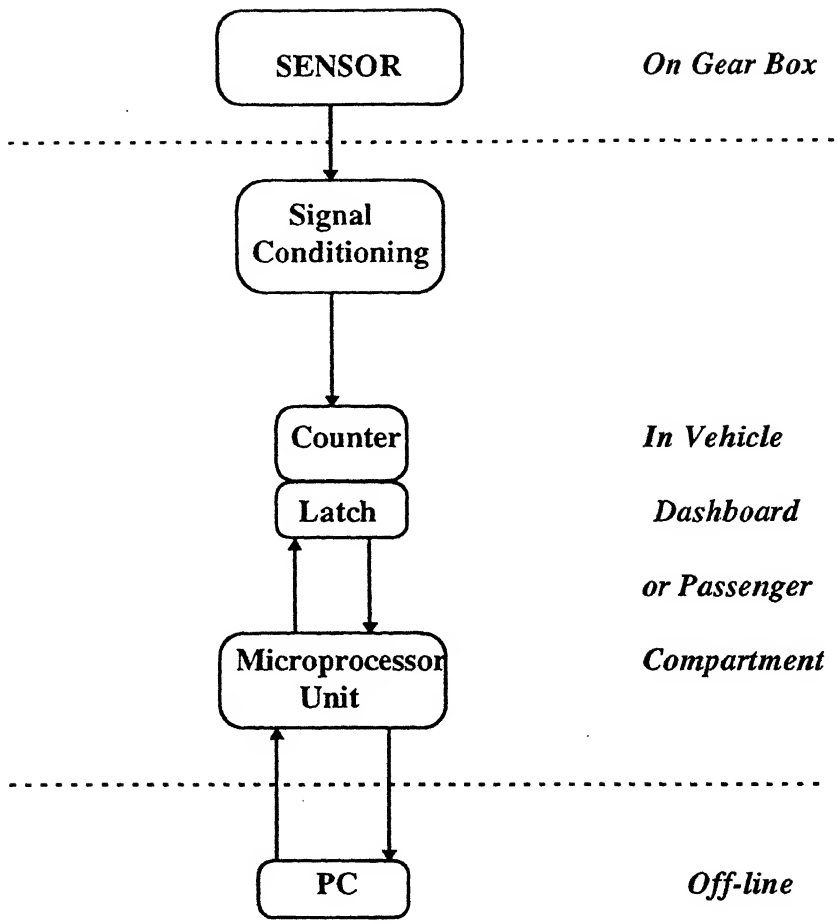

Fig. 4.1 Flow Chart showing the flow of data from the sensors to the PC for analysis

The various components have been described in the following sections.

4.2 Sensors

The sensors (or pickups) are used to sense pulses generated due to the revolution of its distributor. These pulses are then transmitted to the microprocessor for storage. The sensors normally used are of two types:

- Optical sensors and
- Magnetic or Inductive sensors

ITI, Rae Bareilly have shown interest in the work and is likely to undertake the manufacturing of the 'Black Box'. The details of both the types of sensors have been discussed in the following sections along with the diagrammatic representation of the sensors..

4.2.1 Optical Sensors

The optical pickup essentially consists of the following

- circular slotted disc,
- light emitting diode (LED) and
- light sensing diode.

The LED aims a ray of light (usually Infra Red) into the light sensitive diode placed on the other side of the disc. As long as the light sensitive diode can "see" the light, current passes. The distributor housed within the sensor set is connected to the speedometer cable. The distributor rotates with the rotation of the cable. The rotation of the slotted disc in between the two diodes intermittently block the passage of light and triggers the transistor in the light sensitive diode. This intermittent blockage of light results in passage of short pulses of current. These pulses are then transmitted to the microprocessor after undergoing amplification and necessary modifications. A diagrammatic representation of the Optical pickup is shown in Figure 4.2

4.2.2. Magnetic (or Inductive) Sensors

The magnetic pickup essentially consists of the following parts

- magnetic pickup coil and
- reluctor on which tooth are attached.

The wires from the magnetic pickup coil draws the current to the processor which stores it. The magnetic pickups are more popular in automotive ignition circuits, and are thus trade proven.

Here also the distributor is connected to the speedometer cable. This distributor on rotation moves a reluctor past the magnetic pickup coil to produce an electrical impulse, that triggers the transistor in the control box. This transistor perceives the movement of the tooth across the pickup coil as a pulse of current and transmits to the processor after necessary modifications. A diagrammatic representation of the magnetic sensor is shown in Figure 4.3.

For the inductive pickup a pulsar unit from an old autolec/mobilec ignition was salvaged and high voltage circuitry disconnected to make the prototype.

4.3 Relative Advantages and Disadvantages

- Since, the magnetic pickup is drawn from the automotive industry itself, the liability factor is small as compared to the optical sensor which is not very commonly used in the industry.
- The magnetic pickups are more rugged than optical pickups.

In a magnetic pickup, the pulses are dependent on the rate of approach of the susceptor. Hence, below certain speed, this sensor becomes unreliable. However the optical sensor could operate at any speed and give reliable results.

4.4 Location of the Sensors

The best place to mount the pickup is in between the speedometer and the gear box. Also, it cannot be readily built into the gear unit itself. This is because major modifications in the gear box may not be possible at the outset as this would result in high initial cost and also

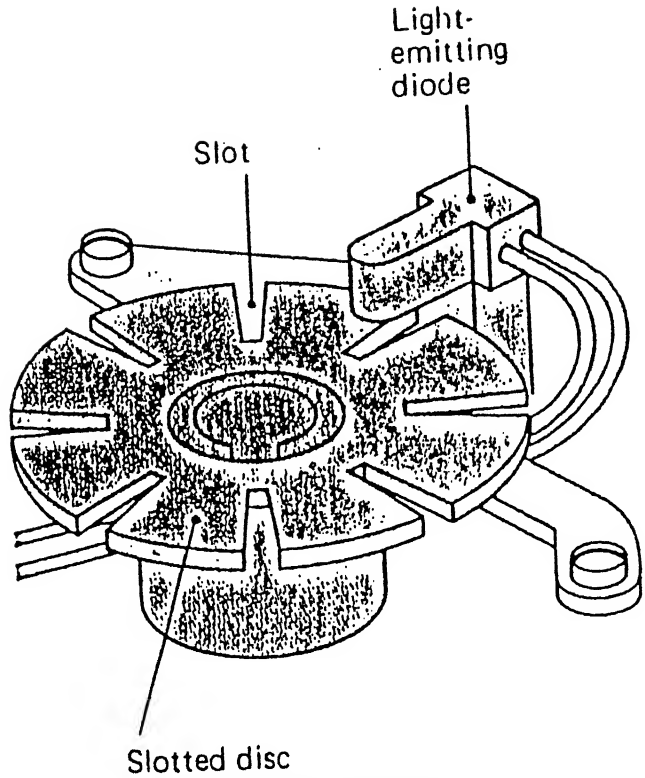

Fig 4.2 Optical Sensor

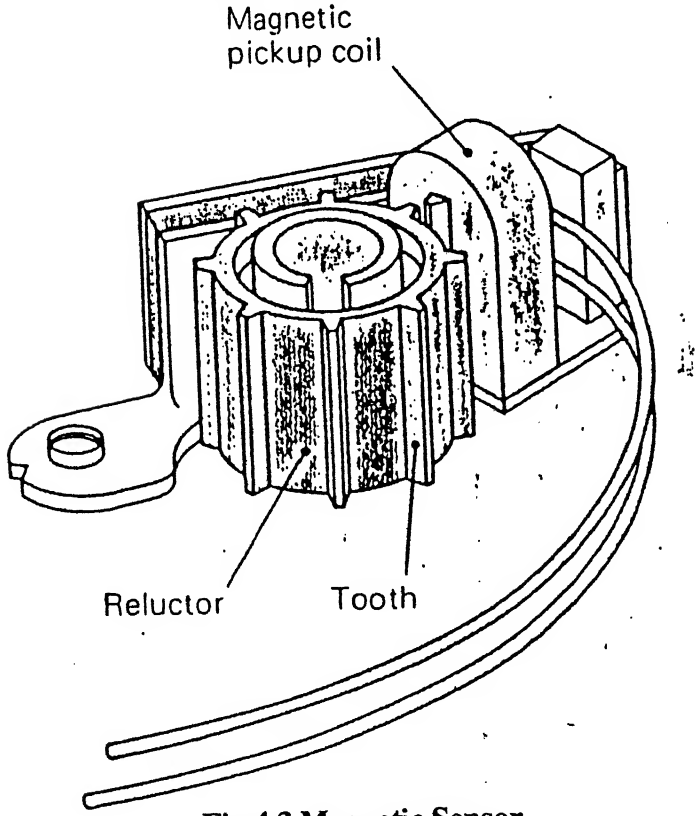

Fig 4.3 Magnetic Sensor

a major change in the design may not be acceptable at all. Hence the proposed location of the sensor is on top of the gear box where the speedometer cable originates. The worm tooth arrangement inside the gear box for the cable is utilized in rotating the sensor. Eventually, it should be possible to replace the speedometer cable totally by the use of these sensors.

4.5 Vehicle Environment and the Sensors

The environment present in the vehicle to which the sensors may be subjected are as follows

- high temperatures (of the order of upto 100 °C)
- high vibrations
- noisy power and
- rough operating environment.

Both the above type of sensors can cope with such adverse environment. The magnetic sensors are already in use in the automotive industry and hence has been tested to withstand such environment. The optical sensors are also known to be rugged and sets with ratings in excess of 100 °C are also available. They are also used in automobile ignition circuits, though the magnetic sensors are more popular.

4.6 Pulses - from Sensors to the Microprocessor

The pulses generated in the pickup unit are transmitted to the microprocessor for storage. These pulses need to be filtered for noise and shaped before feeding to the processor. The pulses are fed into the amplifier unit, to amplify the weaker signals. After amplification, these are fed into the Schmitt trigger for shaping and noise minimization. The amplified and filtered pulses are then fed to the counter and handled by the microprocessor.

4.7 Microprocessor

Inexpensive single board computers are available upto 512k SRAM. These boards have a typical dimension of upto 100mm X 180mm X 20mm and can withstand a temperature of upto 80 °C. Since the board will be housed inside the passenger cabin, this rating is enough.

4.8 Data Volume that can be Stored

Say a typical single board microprocessor discussed above has a memory of 128k of RAM of which 16k is lost for program storage. Hence the available memory space is only 112k. The total data volumes can be calculated as follows for such a microprocessor. Information to be recorded are date, time, distance and fueldata.

			Bytes in ASCII	bytes in Binary
Time in	hhmmss	=>	6	3
Date as	ddmmyy	=>	6	3
Distance (say a maximum of 4Km)		=>	4	2
Max. Speed (say)		=>	2	1
Separator		=>	1	1
Total /unit			19	10

Thus in ASCII 112k corresponds to 6036 units whereas in BINARY 112k corresponds to 11464 units.

Thus on a daily basis in ASCII the resolution of samples that can be obtained is 4 samples/min and on weekly basis a resolution of 0.6 samples/min can be obtained. Similarly in BINARY on a daily basis a resolution of 8 samples/min and on a weekly basis 1.14 samples/min can be obtained.

The resolution calculated are in the worst cases. Thus the resolution obtained both in ASCII and BINARY units are so small that a detailed data set can be obtained and this further facilitates exhaustive analysis.

The above analysis was done considering the point of view of storage.

If the frequency of pulse acquired by the sensors is considered then the calculations are as follows. The Maruti 800 has been used to give an example calculation.
The final drive ratio of Maruti 800 is 4.351. In other words, 4.351 rotations of propeller shaft results in 1 revolution of the wheel. The wheel has 5.65" width (the tyres are 5.65" wide with an 80% aspect ratio) and 12" rim. Thus diameter = 12+2*0.8*5.65 = 21.04".
Thus the wheel circumference is 66.099", and hence the distance covered by the wheel in one revolution is 1678.91mm.
If the vehicle travels at 100Kmph, the wheel rotates at 992.7 rpm. Hence the prop shaft rotation is 992.7*4.351 = 4319.25 rpm.
The speedometer gearing in Maruti 800 is 5 : 18 and if the frequency of pulses allowed is 8 pulses/rev, this amounts to a pulse frequency of 159.97Hz on the sensor. Thus a pulse rate of 200Hz is more than adequate for any condition and such rates are easily achieved.

In case of Premier 118 NE cars traveling at 150Kmph the pulse frequency is 222Hz.

4.9 Options Available for Use in Future

Apart from the basic data like time, distance, fuel attributes, temperature etc., other sensors and parameters such as accelerometer, smart cards etc., may be added for use in the future. The flag which is the first byte recorded by the processor can be used for driver identification. Whenever the driver in the vehicle changes, he inserts his smart card into the mechanism and his identification is done. This continues until the driver changes duty, i.e., the next driver inserts his smart card. Thus drivers could be identified and analysis carried out for each of the drivers separately.

Chapter 5

SOFTWARE DESIGN

5.1 Salient Features of the Software

The software was developed in two stages as follows:

1. Basic program logic written in C programming language and
2. Graphical representation of analyzed data in the Graphical User Interface Panel.

A stream of data is acquired from the 'black box' using the RS-232 serial ports available in the PC. This data is stored in a file which is to be accessed for further analysis. Data such as time, distance, fuel consumed, temperature etc., are initially in the form of pulses of definite time interval. This data needs to be integrated to give cumulative data in the form of time, distance, fuel, temperature coordinates. This step is a preprocessor to the project actually built.

The cumulative data of time and distance is the input for the present work. Since real time data was unavailable, fuel consumption and temperature details could not be analyzed. However, the logic of their analysis is also similar to the work described below. The data was obtained from a VTI Simulation Model for narrow roads and heterogeneous traffic conditions. Traffic was generated for an hour with a flow of 200 vehicles per hour and the required data statistics of one particular vehicle was obtained in the required format.

5.2 Flow Chart of the Software

The flow of data across the software and the logic behind its analysis are diagrammatically represented by the Flow Chart shown in Figure 5.1.

As shown in the flow chart, data is first read in the form of time and distance coordinates. This data is further used throughout the program for analysis and graphical representation. The number of points to be analyzed in each iteration can be controlled from within the program. This was done to facilitate analysis of a large set of data points by dividing it into smaller blocks, so as to minimize the array size of the variables used. Thus, the detailed vehicle movement data for fairly large amount of time can be easily analyzed. Using the coordinates of time and distance, the Time total and Distance total is determined easily since the data itself is in cumulative form. A polynomial curve for time Vs distance is fitted to obtain the equation of distance in terms of time for further analysis. The curve fitting was done only for data where the vehicle is moving. Hence a piece-wise curve fitting had to be resorted to. Using the first and second differential of the equation, velocity and acceleration at corresponding points are determined. Velocity and acceleration data are further manipulated to give parameters such as average velocity, maximum velocity, etc. A detailed analysis of a known number of points before every halt is made using a very small interval of time (one second), which can be controlled. The period where the acceleration (and hence the velocity) changes/varies abnormally, are also analyzed. The length of such period (or the number of points in the vicinity of this period) can be controlled programatically.

With the basic analysis done, these analyzed data are graphically displayed using the Graphical User Interface Panel. This panel contains graphs, strip charts, numeric display boxes, indicators and control buttons. Since plotting graph for every set of analyzed data causes repeated panel loading, it results in memory overflow. Hence the analyzed data was plotted on the graph (i.e., velocity Vs time) after reading and analyzing all the points.

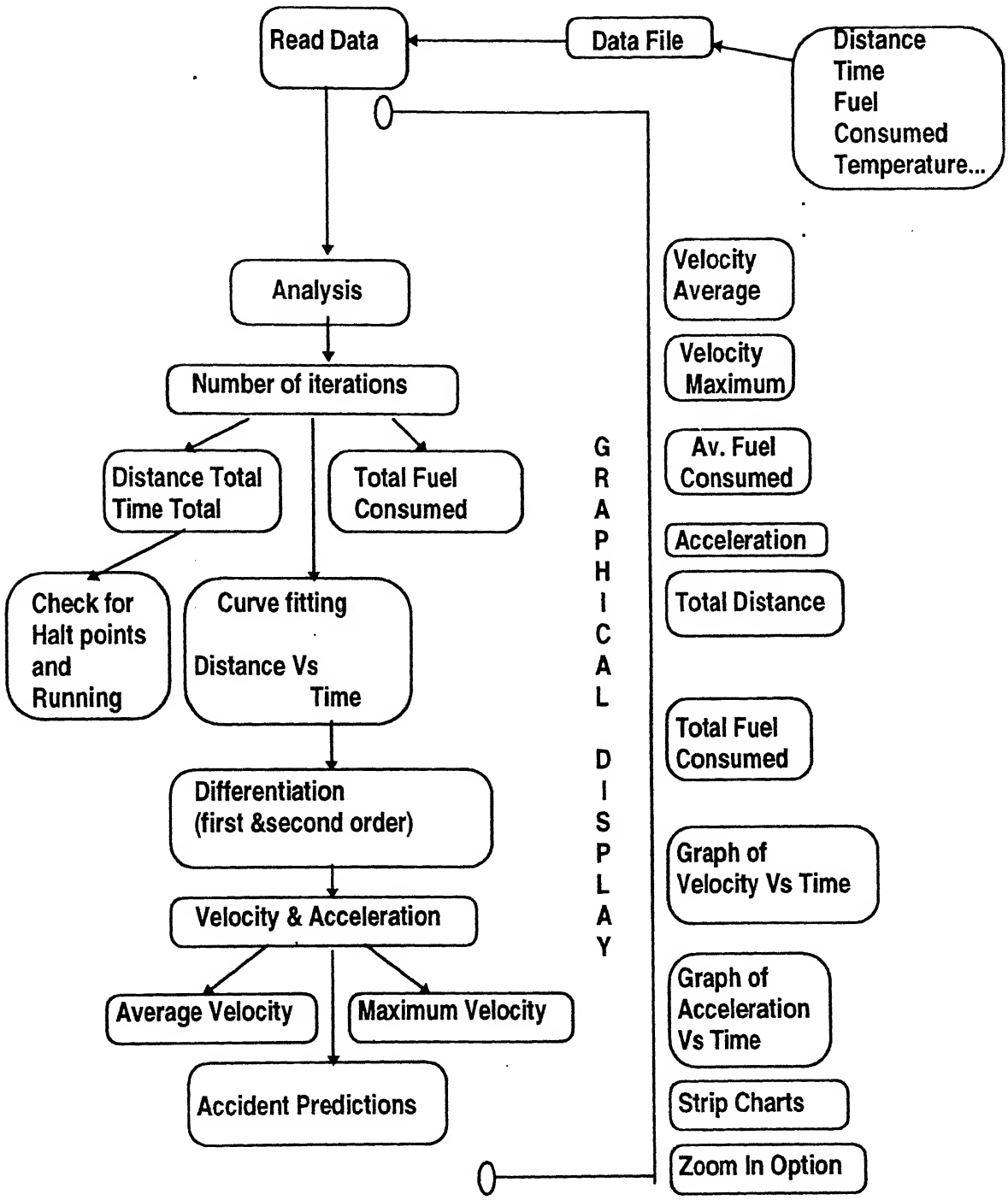

Fig.5.1 Flow Chart of Project

5.3 Description of Various Program Modules.

The modules are divided into two broad categories as follows:

1. Modules for data analysis and
2. Modules for Graphical Display.

The description of each of these modules along with their flow charts are detailed below.

5.3.1 Data Analysis Modules

The whole software is divided into small modules to facilitate easy understanding of the program and also to enable easier editing of the program. Some of the important modules have been discussed in the following sections.

5.3.1.1 Module to Read Data

The first of these modules consists of function to read the data in terms of time and distance coordinates from the preprocessed stored file. The number of points that are to be read can be controlled in the function. This was done mainly to enable the analysis of a large data of fairly large time period. The flow chart of this module as shown in Figure 5.2, is self explanatory.

5.3.1.2 Module to Obtain Halt Points and Running Points Attribute

Successive data points are compared in this function. Whenever the program encounters two successive distance coordinate as equal, this module is activated. The first time of halt (i.e. time when the vehicle has just halted) and the corresponding distance coordinate, the last time of halt (i.e. time just before the vehicle starts to move again) are stored. If there are a number of halt points, a number of such start and end points are stored. From these

stored points, duration of vehicle halt and the corresponding halt coordinate(distance) is obtained. This data is appended into a separate file. Whereas for the successive distance

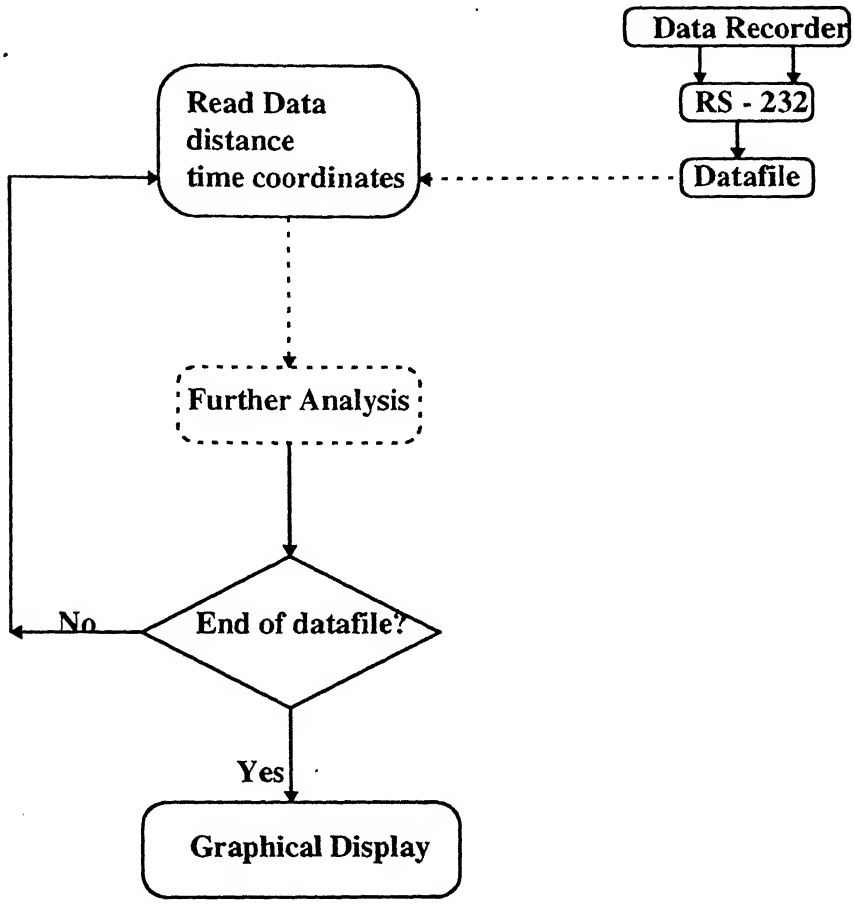

Fig. 5.2 Flow Chart of the Module to Read Data

coordinate not being equal, the module for running points attribute is activated until a halt is encountered. Unlike the halt module, each of these points are stored to be fed into the Curve fit module. The flow chart of this module is shown in Figure 5.3.

5.3.1.3 Curve Fitting

Since the main objective of this program was to obtain the velocity and acceleration, a curve was fit for Distance Vs Time data. This was done essentially to obtain the

coefficients of distance-time equation, so that velocity and acceleration equations can be obtained by differentiation of this equation. Now the minute analysis of velocity and

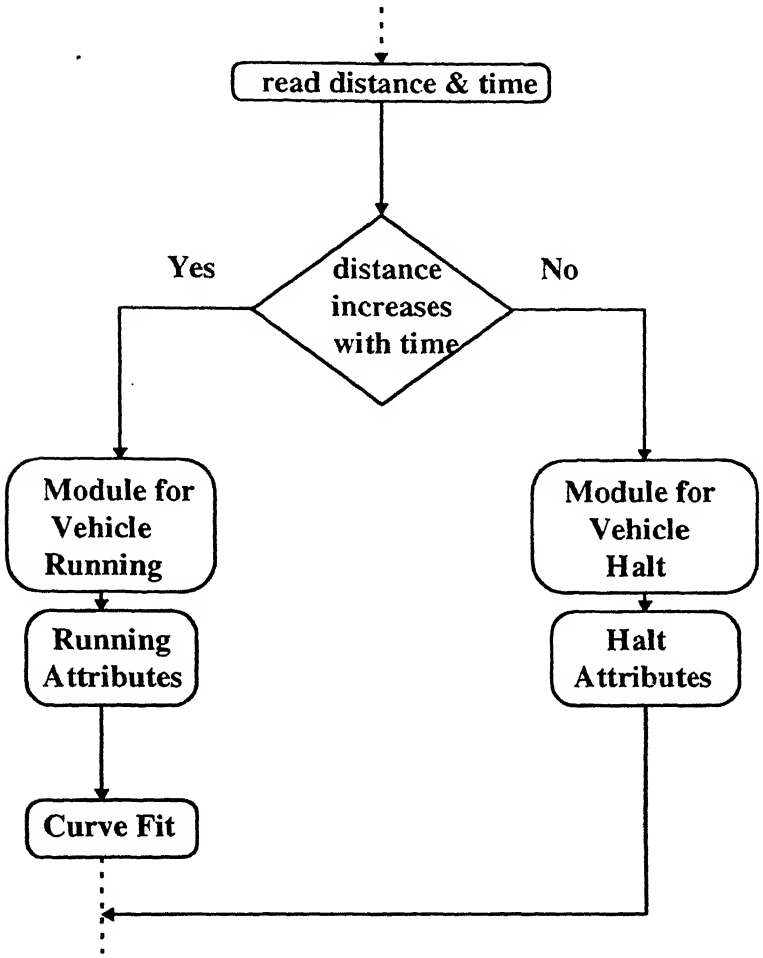

Fig. 5.3 Flow Chart of Module to obtain Halt Points and Running Points Attribute

acceleration is accomplished using these two equations. The curve was fit on only those points where the vehicle was running. The following equation was fit to the data:

$$d(t) = a_o + a_1*t + a_2*t^2 + a_3*t^3 + ...+a_n*t^n.$$

where $d(t)$ represents distance in terms of time, t

a_o, a_1, a_2, ..., a_n are the coefficients of the polynomial equation of order n.

This equation is obtained for each and every running point. To fit this curve, a built in library function 'PolyFit' from the Advanced Analysis Library of LabWindows/CVI was

made use of. This function finds the coefficients that best represent the polynomial equation to the data set. The i-th element of the output array is obtained using the following formula:

$$z(i) = \sum_{j=0}^{order} Coeff(j) * | t(i)^j |$$

where z(i) represents the calculated distances,
Coeff(j) represents the coefficients obtained after curve fitting,
order is the order of the polynomial and
t(i) represents the time.

The mean squared error (mse) is obtained using the following equation:

$$mse = \sum_{i=0}^{n-1} | z(i) - d(i) |^2 / n$$

where z(i) is the calculated distance and d(i) is the actual distance coordinates. Here n is the number of sample points.

The order of the polynomial used was 10 (i.e., 10th degree polynomial equation). However, if the number of running points are less than 10, the order used is one less to the number of points in that particular run.

5.3.1.4 Module to Calculate Velocity and Acceleration

The equation obtained from the last function is differentiated twice to obtain the velocity and acceleration at corresponding distance-time coordinate set. The first differential gives Velocity coordinate and the second differential results in corresponding acceleration. This equation was used only on the points where the vehicle was running.

5.3.1.5 Module for Detailed Analysis - Halt Analysis

To facilitate an easy understanding of the drivers behavior and also to point to possible cause of accidents, if any, a module for a detailed analysis of certain important regions of

the curve has been included. This module is activated whenever the vehicle comes to a stop and also when the variation in its velocity (hence acceleration) are large. The number of points to be analyzed depends on the degree of detail required. Halt analysis is included mainly to analyze the response of the driver to different possible situations and most importantly to gauge the possible causes of accidents and the reactions of the driver in the short span of time when the accident occurs. The output of each of these modules are printed in separate files. This facilitates further study and plotting.

5.3.2 Modules for Graphical Display

The built in User Interface Editor of the LabWindows/CVI was used to generate graphical display of the results. The editor has been explained in Chapter 3 in detail. The control events predominantly used were EVENT_COMMIT and EVENT_RIGHT_CLICK. Event COMMIT is returned whenever the control button is activated either by the left button click of the mouse or by using the respective shortcut key assigned from the keyboard. Similarly event RIGHT_CLICK is returned when either the right mouse button is clicked or the assigned shortcut key is pressed. In both these cases the control button in the Graphical Display panel should be in 'HOT' mode (explained in Chapter3). By clicking on the control button the callback function assigned to it is activated. The callback function to these controls have also been coded in C programming language. The following callback function module have been coded:

5.3.2.1 Module for Graph Plot

The object used in this callback function is the Graph. This object has been assigned the 'HOT' mode because manipulations have to be done on the Graph, on the screen itself, to facilitate ZoomIn control. However, if this facility is not required, the object maybe a simple indicator mode. To plot the Graph a control button 'PLOT' has been displayed on the Panel. When this control button is committed (i.e., event COMMIT), either using the left mouse button or menu key **'Ctrl+p'** from the keyboard, the callback function assigned

to this control gets activated and using the processed data assigned to it, passes the information to the PlotXY function of the User Interface Library. The PlotXY function returns the Graph which is displayed on the screen. In the present case Velocity Vs Time has been plotted. To help in identifying the purpose of the PLOT button control, a PLOT AID popup window may be activated if the right mouse button is clicked to return event RIGHT_CLICK. The average velocity and the maximum velocity are also displayed in the numeric display box on the Panel when this control is activated. The flow chart of the module is shown in Figure 5.4 below.

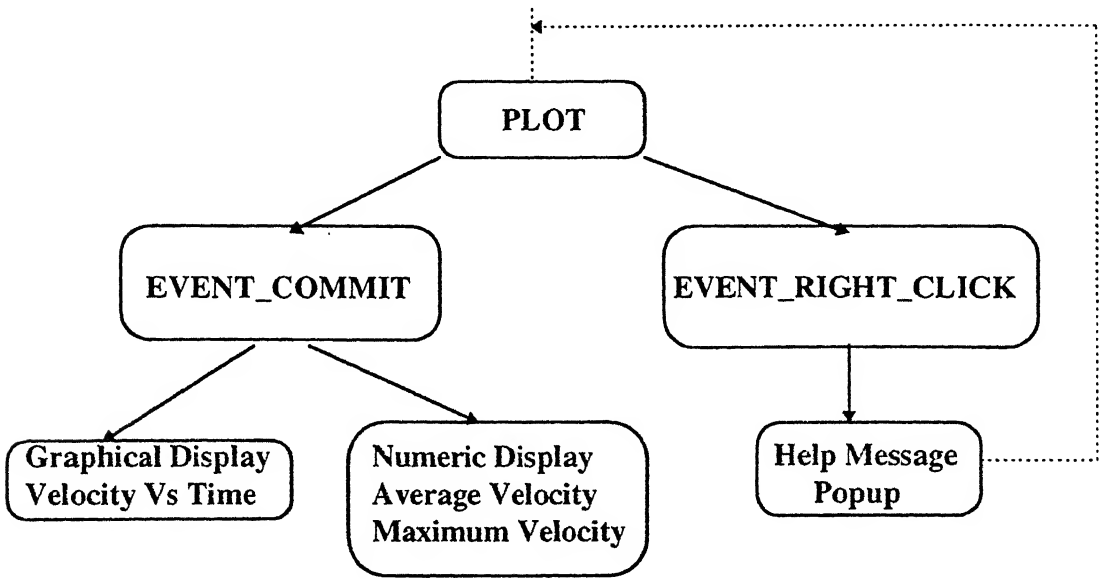

Fig. 5.4 Flow Chart of the Module for Graph Plot

5.3.2.2 Module for Zooming In

This function has been included to facilitate a detailed and closer look at the speed characteristics of the vehicle. A 'ZOOM' control button is provided on the Panel. When this button is activated, the module asks for two points between which Zooming is to be done. The control maybe activated either by the left mouse button click or by the shortcut key **'Ctrl+z'** from the keyboard. The option to Zoom is available only after the Graph has

been plotted. The two required points can be selected using the 'Long Cursors' provided on the Graph. These cursors can be moved on the Graph with the help of the mouse. No shortcut keys are assigned to it. The values pointed to by the cursors act as inputs to the function. This enables the function to set fresh axis ranges. A SetAxisRange function available in the LabWindows/CVI User Interface Library, is made use of to set new axis range once the two points are read. Both the 'x' axis and the 'y' axis can be set using this built in function. However since the velocity is limited to a relatively very small value when compared to time, only the 'x' axis has been made to vary using the said built in function. The callback function associated with the Zoom control button, replots the graph between the two selected points to a smaller scale. A third cursor (with Short Cross) available on the graph may be utilized to read the points where it is pointed to. This facilitates readability of even those intermediate points where data is unavailable due to the 'pulse time-interval'. The selected data are displayed as numeric display on the Panel. The flow diagram of this module has been shown in Figure 5.5.

5.3.2.3 Module for Zooming Out

The graph after having been replotted to a smaller scale in the last module is to be restored back to its original plot. This function restores the plot to its original shape, in other words it Zooms Out. This is accomplished by plotting the graph after setting the axes to the original scale. Like the last module, use of the library function SetAxisRange has been made to restore the scale to the original plot. This module is activated by the left mouse button click or by pressing **'Ctrl+r'** from the keyboard. The restore plot module works only if the graph has been zoomed in.

5.3.2.4 Module for Printing the Graph

The 'PRINT' button specified in the Graphical Interface Panel, is the control button for this module. Left mouse click on this button or **'Ctrl+l'** activates this module. When this is done a message is displayed in the Message Popup. The message gives information

about the file into which the graph is stored, and also how it can be printed on paper. The module can also be used to configure the printer and print the graph directly avoiding the intermediate step of storing in separate file.

5.3.2.5 Module to Quit from the Program

After all the points are analyzed, a Quit option has been provided to quit from the program. This can be achieved by activating the 'QUIT' button on the panel. The button may be activated by the left button click of the mouse or by pressing **'Ctrl+Esc'** from the keyboard. Once this button is activated, a Confirm Popup window appears providing option to view the acceleration Vs time graph plot. Two buttons have been provided on this window, 'NO' and 'YES' buttons. When the Yes button is clicked, a XY Graph popup appears showing the plot of acceleration in the Y axis and time in the X axis. two more buttons have been provided on this popup, the 'OK' and the "Print' buttons. The print button when activated, prints only the XY popup graph directly into the printer. After printing, the callback 'Shutdown' is activated and the program is terminated. However, when the OK button is selected, the program is terminated without printing the contents of the XY popup. The flow chart for this module is illustrated in Figure 5.6.

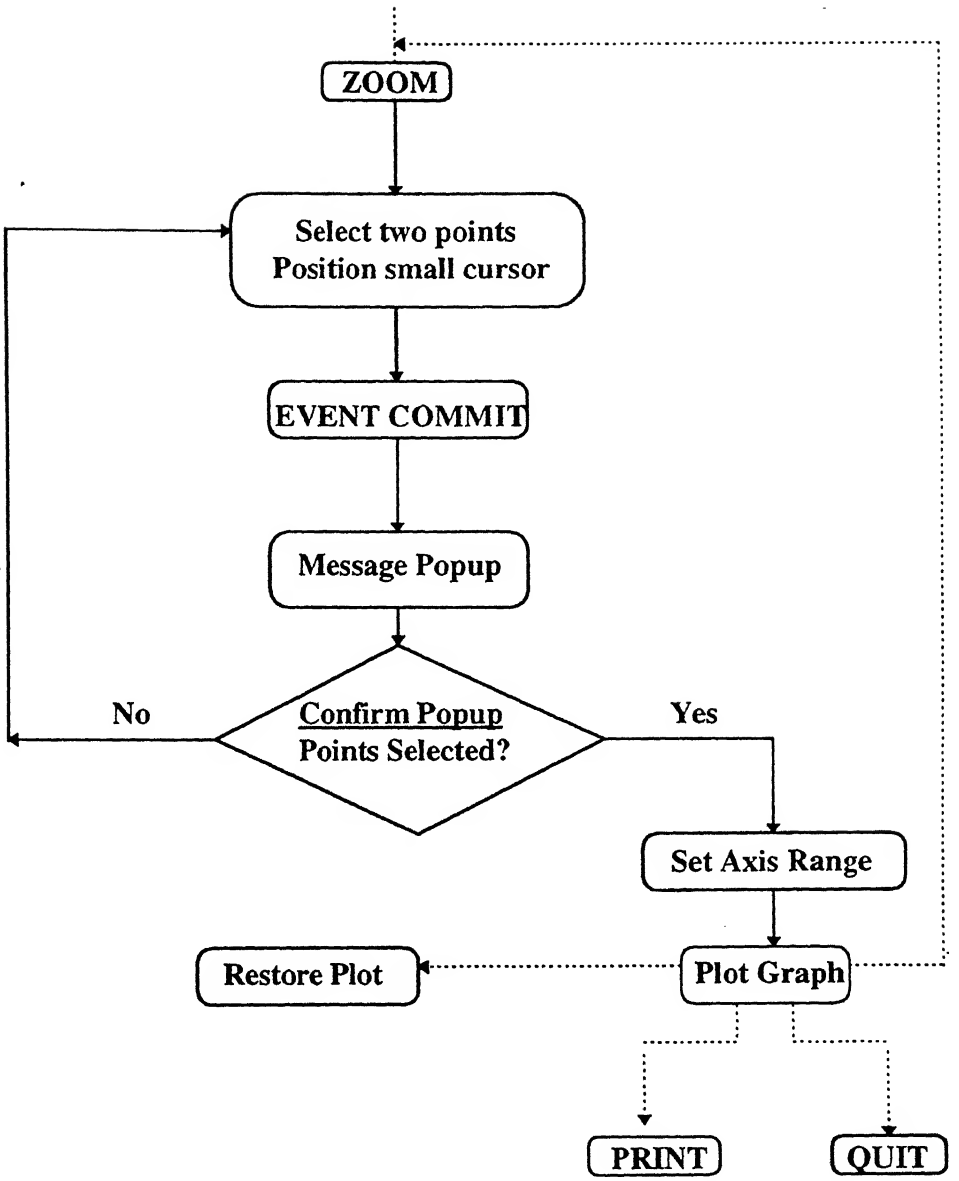

Fig. 5.5 Flow Chart of the Module for Zooming In

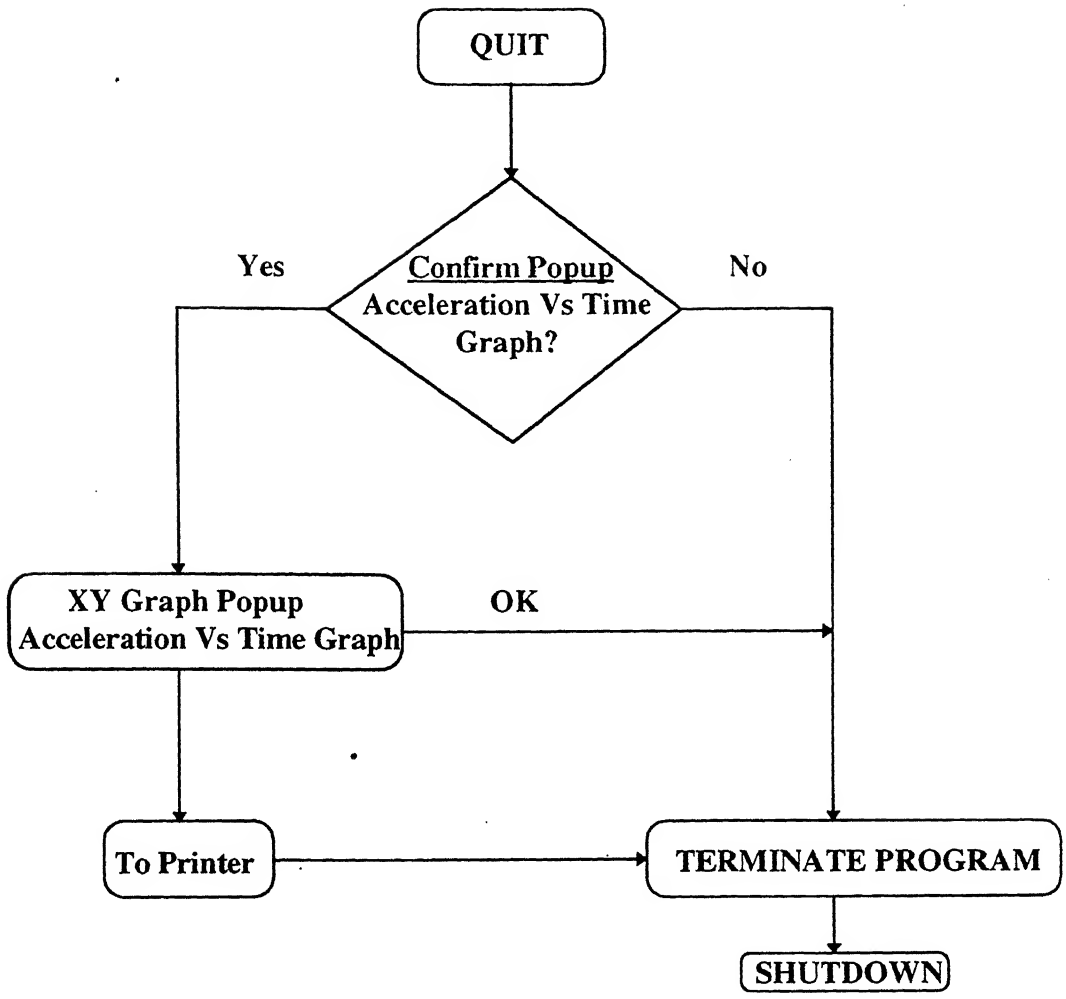

Fig. 5.6 Flow Chart of the Module to Quit

Chapter 6

DISCUSSIONS OF RESULTS, CONCLUSIONS AND SCOPE FOR FUTURE WORK

6.1 Discussions

A vehicle on a road stretch may have to interact purely with road or with road and traffic. Based on the interactions, different situations can emerge for the driver. If a driver is allowed a free stretch of road without any hindrances, he would move at his desired speed. However, due to influences of surrounding vehicles and also the road conditions, among other factors, the driver deviates from his desired speed. The interaction of the driver with other vehicles may be in many ways. For example, while overtaking, if the vehicle ahead does not move to the left and slow down, or if the oncoming vehicle is fast approaching, the driver is resigned to follow the vehicle ahead until he gets an opportunity to overtake. In such circumstances the following vehicle slows down to the speed of the vehicle ahead. When the driver starts overtaking, he accelerates rapidly till the overtaking maneuver is accomplished. Similarly, when other vehicles overtake, the driver of the overtaken vehicle is influenced by their moves. The road conditions play a vital role in the behavior of drivers while driving. On inhibitive road stretches the driver has to travel below his desired speed. If the driver doesn't do so, it leads to excessive fuel consumption, wear and tear and excessive use of brake which results in heating of the axle. This process also eventually results in accidents. The axle may also get heated when the tyres fail. Failure of tyres at high speeds lead to major accidents. Sometimes due to sudden application of brakes, the wheel gets locked. This results in skidding or overturning as at curves. In such cases there is no rotational movement of the wheel. In winters at high altitudes, roads are layered with thin sheet of ice. Overspeeding on such roads also lead to serious accidents. The effect of power weight ratio is very important in diagnosing accident causes. A vehicle traveling at nearly its full power and trying to overtake, requires a larger

overtaking sight distance due to its inability to accelerate rapidly as compared to a vehicle moving at a slower speed. Many head-on collisions occur due to inexperience and erroneous judgment of the drivers. It is easier for vehicles traveling at lower speeds to accelerate than for vehicles already traveling at a high speed.

A few of the various situations that a driver may face has been discussed above. This should further aid in inferring about the travel characteristics of the driver from the results of the analysis of the sample data.

6.1.1 Discussions of Result

Since real time data was unavailable, the required data was simulated using the VTI model. The simulation was carried out for a period of 1 hour and for a road stretch of 5Km. The characteristics of the road stretch considered were as follows:

- Intermediate Lane Road, 5.5m wide.
- Hard Shoulders of 2m width present on either side.
- The traffic flow considered was 200veh/hr.
- Traffic consists of Maruti cars and Trucks.

The data for one vehicle (in this case a Maruti car) was sorted out from the file generated after simulation. The data in terms of time and distance coordinates were used in the sample analysis. Further to simulate actual journey, halts were added at the end and also at intermediate sections.

1. Result of the Analysis of Data Set 1

The output of the analysis is tabulated in Tables A-1 and A-2 in the Appendix. The data represents a part of the journey before the vehicle halted. The velocity for the entire journey is close to the average value without much deviations. The acceleration also suggests that the journey was more or less uniform. However, the deceleration was rapid just before the halt suggesting that the driver applied brakes to stop the vehicle rapidly. In

the last 4.5 seconds the vehicle was brought to rest from a speed of about 28Kmph. Further a look at the data obtained after detailed analysis (Table A-2) suggests that the deceleration was rapid.

It can be deduced that the road stretch was fairly uniform with moderate traffic. It can also be inferred that the driver of the vehicle was traveling at a uniform speed along this road stretch. However, the last few seconds before halt suggests that the driver had to apply brakes suddenly, probably on encountering an obstruction. Since the vehicle was decelerating for sometime before it halted suddenly, it may be concluded that the obstruction could be another vehicle traveling ahead. Probably the vehicle ahead came to a stop abruptly which forced the driver of the following vehicle to do the same.

In the light of the discussions above, it can be concluded that the vehicle was being driven at a uniform speed for the major part of the journey. The driver had to stop the vehicle suddenly although the vehicle was decelerated much earlier due to obstruction in the form of another vehicle ahead.

2. Result of the Analysis of Data Set 2

The vehicle halted at an intermediate point during its journey. The set of data which includes this halt was analyzed. A halt at the end was introduced to simulate actual journey. The analysis results have been tabulated in Table A-3 and A-4. the vehicle first halted at time coordinate of 145.06s and at distance coordinate of 1777.79m. The detailed analysis suggests that the halt was uniform. It can be observed that the velocity and acceleration before the first halt were fairly uniform. The halt at the end of the journey was also uniform. It can be observed that the velocity and acceleration before the first halt were fairly uniform. However, brakes were applied so as to stop the vehicle traveling at about 31Kmph in only 6s. It is also observed that the halt was for 8s. This suggests that the driver had to stop the vehicle because of some obstruction ahead. the obstruction could have been in the form of a parked vehicle or a signalized intersection or a railway

crossing etc. The vehicle may have stopped at a tollgate. The vehicle speeds were not consistent after the first halt. The values near the last halt suggests that this halt was of routine manner without rapid deceleration.

The discussions above suggests that the first stretch (before first halt) of the journey was uniform. The second part of the journey suggests the inconsistency of the driver. This could possibly be due to a different traffic and road conditions.

3. Result of the Analysis of Data Set 3

The result of analysis of the data set 3 have been tabulated in Table A-5 and A-6 in the appendix. From the table it can be seen that the driver in this case had been traveling at a higher velocity before the analysis was done. This can be said because the corresponding acceleration values are negative indicating deceleration. After decelerating, the vehicle accelerates rapidly and finally comes to an abrupt halt from a velocity of 72.25 Kmph in only about 3 seconds.

The most likely inference drawn from this is that the driver of the vehicle must have spotted a vehicle ahead and unable to perform "flying overtaking" slows down and follows the vehicle ahead. When the driver had an opportunity to overtake, he accelerated rapidly and was in the process of overtaking when the vehicle comes to a sudden stop. This clearly suggests that an accident had occurred. Due to the nature of the travel, it probably was a head-on collision. It can be inferred that the accident might have occurred due to the negligence of the driver of the overtaking vehicle or unavailability of the required sight distance. It could well be the case of wheel locking as a result of sudden application of brakes. From the values in Table A-6 it can be easily deduced that the vehicle was involved in an accident.

In all the above three cases discussed merely observing the values in the table does not provide the actual feel of the field situation. However, when graph is plotted in the GUI,

the actual situation could be easily visualized and analyzed in detail using the Zoom controls. A detailed analysis of data such as fuel consumed, temperature changes etc., along with time and distance analysis could be helpful in bringing out a complete diagnosis and also highlight the causes of accidents. For example excessive use of brakes leads to high fuel consumption and inefficacy of the engines. The first sign of tyre failure is heating up of the axle. This information could also be used to warn the drivers in the vehicle itself. Thus only the analysis of time and distance data does not lead to concrete diagnosis.

6.2 Conclusions

Design of a software for the analysis of detailed vehicle movement data, with Graphical User Interface has been achieved. The software is PC based and user friendly.

Data is to be obtained in the form of pulses from sensors fitted to the vehicle. The pulses are transferred to the microprocessor after necessary amplification and modifications. The microprocessor stores data for a given time frame which depends upon its memory. This data is to be downloaded into the PC using RS-232 ports.

The program has been divided into modules, each of them being self explanatory. To make the software user friendly, a Graphical User Interface has been incorporated using LabWindows/CVI as the software developing tool. Various control buttons help in using the software effectively.

This software analyses the data and displays it in the form of graphs, strip charts and numeric controls. Zoom options have been added to view the journey details on the graph. A module for detailed analysis of halt points has also been incorporated to obtain a micro analysis of data before halts. The analyzed data is stored and can be viewed or printed. The analysis of detailed vehicle movement data helps in studying the behavior of the drivers to different situations on the road. Sample data have been analyzed to show the utility of the software.

6.3 Suggestions for Future Developments

The present work is the nucleus around which a more versatile and user friendly software can be developed. The areas which may be expanded to give the required results are as follows:

- Adding subroutines for fuel consumption and temperature characteristics analysis.
  A mere analysis of time and distance data cannot be used to predict the driver's behavior. A knowledge of corresponding fuel consumption and also the temperature data could further help in diagnosing the causes of accidents.
- Load sensors could be added and the function for the analysis of the data obtained from them could be included. This would help in satisfying the needs of transport operators.
- Options for the analysis of specific regions can be included in the program. Detailed analysis of points selected from the graph should be incorporated as this would make the software more readily usable.
- Relative correlation between different parameters could be looked into. The present work looks only at the time Vs velocity and acceleration attributes.
- The instrumentation part of the project could be enhanced with the background discussed above.
- Some of the road and traffic attributes on which the vehicle traveled can also be inputted so as to make reasonably good predictions about the capabilities of the driver and cause of the accident, if any.

APPENDIX

Table A-1. Output of Analysis of Data Set 1.

Time (sec)	Distance (m)	Velocity (Kmph)	Acceleration (Kmph/s)
37.8300	335.8400	43.2360	0.191880
47.7800	460.5100	44.5320	0.065880
50.9700	500.4800	44.6760	0.030960
60.2700	620.6400	44.5680	-0.047520
66.4600	700.6100	44.1720	-0.077400
72.3100	774.9600	43.6680	-0.089640
74.3700	798.9000	43.4880	-0.090720
82.9700	898.8300	42.7320	-0.078480
96.3100	1061.9600	42.0120	-0.025920
99.5100	1097.1600	41.9400	-0.010800
115.9000	1217.1700	42.3720	0.059400
117.6900	1297.1400	42.5160	0.064800
127.3500	1419.8200	43.2360	0.084600
133.6600	1499.9600	43.7760	0.086400
143.1100	1619.7900	44.5320	0.072720
149.4200	1699.8000	44.9640	0.053640
158.7500	1819.8800	45.2880	0.013680
164.9600	1899.8000	45.2520	-0.018360
168.4000	1943.5200	45.1800	-0.037080
171.4300	1977.0600	45.0360	-0.054000
182.2700	2097.0600	44.1000	-0.112680
193.1100	2217.0600	42.6240	-0.161280
196.7200	2257.0200	42.0120	-0.173520
200.0800	2299.4900	41.4000	-0.183240
203.3500	2336.5400	40.7880	-0.190080
205.3700	2358.6600	40.4280	-0.193680
208.3500	2391.2900	39.8160	-0.196920
214.5800	2458.6400	38.5920	-0.197280
218.2800	2498.6400	37.8720	-0.193680
229.5200	2618.6800	35.8560	-0.164160
236.9600	2698.5800	34.7400	-0.131040
237.7200	2706.7500	34.6320	-0.127080
238.6000	2716.0000	34.5240	-0.122400
239.2600	2722.8400	34.4520	-0.118800

249.1400	2825.4900	33.5520	-0.060120
253.3800	2869.5400	33.3720	-0.032400
255.7700	2890.7400	33.3000	-0.016560
257.5700	2906.7100	33.3000	-0.004680
258.1300	2911.6700	33.2640	-0.001080
259.2200	2921.3400	33.3000	0.006480
259.8500	2926.9300	33.3000	0.010440
260.3200	2931.1000	33.3000	0.013680
261.9000	2945.1100	33.3360	0.024120
263.4000	2958.0400	33.3720	0.033840
263.9800	2962.9900	33.3720	0.037440
264.5700	2968.2000	33.4080	0.041040
269.5700	3020.1500	33.6960	0.072000
272.8400	3047.0600	33.9480	0.090720
273.6400	3053.4600	34.0200	0.095040
280.2500	3093.5900	34.7760	0.127080
282.4400	3106.8800	35.0640	0.136080
283.2600	3111.8600	35.1720	0.139320
284.7300	3120.7800	35.3880	0.144360
285.5500	3126.5000	35.4960	0.146880
285.8700	3128.8200	35.5680	0.147960
290.2700	3175.6000	36.2160	0.159480
298.4400	3275.5200	37.5840	0.168120
299.3000	3285.7100	37.7280	0.167760
300.8400	3302.9700	37.9800	0.167040
301.7700	3313.0400	38.1240	0.166680
302.1200	3316.9000	38.1960	0.166320
311.7100	3436.8700	39.7080	0.145440
313.0800	3453.1600	39.8880	0.140760
313.5000	3458.0700	39.9600	0.138960
314.1400	3465.3700	40.0680	0.136800
315.5900	3483.0000	40.2480	0.130680
318.7600	3519.9600	40.6440	0.116640
320.5500	3540.5500	40.8240	0.108000
321.7900	3554.6700	40.9680	0.101520
322.8400	3566.5400	41.0760	0.095760
329.5700	3641.5800	41.5800	0.056880
336.7400	3721.5200	41.8320	0.012240
337.2600	3727.0300	41.8320	0.009000
338.6000	3741.2200	41.8320	0.001080
339.0700	3746.2000	41.8320	-0.001800
340.5600	3761.9800	41.8320	-0.010800
340.6200	3762.6100	41.8320	-0.011160

341.9900	3777.1200	41.8320	-0.019080
342.4600	3782.1000	41.7960	-0.021960
343.1000	3788.9400	41.7960	-0.025560
349.4500	3862.7900	41.5080	-0.057600
352.4600	3897.8000	41.3280	-0.069840
354.7300	3921.3800	41.1480	-0.077400
366.2800	4041.3900	40.1400	-0.088560
375.9100	4141.4400	39.4200	-0.056520
383.6800	4240.5100	39.1680	-0.004680
385.5700	4265.5300	39.1680	0.010800
385.9500	4270.3900	39.2040	0.014040
387.3000	4285.6400	39.2040	0.025200
387.7400	4290.4000	39.2400	0.029160
387.8700	4291.7900	39.2400	0.030240
388.9400	4303.2200	39.2760	0.039240
390.0400	4314.9700	39.3120	0.048960
391.1500	4326.8200	39.3840	0.058680
391.6100	4331.7400	39.4200	0.062640
392.7000	4343.3800	39.4920	0.072000
394.1300	4358.6500	39.6000	0.083880
395.0700	4368.6900	39.6720	0.091800
396.7800	4386.6100	39.8520	0.104760
403.0300	4465.3600	40.6080	0.139680
404.9700	4490.6400	40.8960	0.143640
408.2900	4527.1600	41.3640	0.139680
417.3900	4627.2600	42.2280	0.019800
418.2700	4636.9400	42.2280	-0.003600
420.0200	4656.0100	42.1920	-0.057600
420.4800	4661.0100	42.1560	-0.073800
421.3600	4670.5400	42.0840	-0.106920
422.2900	4680.5500	41.9760	-0.145080
423.8400	4697.5900	41.6880	-0.217440
427.2900	4735.5000	40.6080	-0.420120
428.3900	4747.5900	40.1040	-0.498240
431.5700	4782.5400	38.1240	-0.766440
433.9100	4807.5300	36.0360	-1.308720
436.2600	4832.6300	33.3360	-1.495280
439.5300	4867.5500	28.3320	-2.136240
441.5300	4887.7900	18.0124	-3.971480
443.0225	4895.6800	06.4995	-6.104000
443.9800	4897.9870	0.0000	0.000000

Table A-2. Detailed Analysis of Data Set 1 at Time Interval of One Second

Time (sec)	Velocity (Kmph)	Acceleration (Kmph/s)
391.0225	39.3676	0.057600
392.0225	39.4295	0.066240
393.0225	39.4999	0.074880
394.0225	39.5788	0.083160
395.0225	39.6660	0.091080
396.0225	39.7612	0.099000
397.0225	39.8641	0.106560
398.0225	39.9743	0.113760
399.0225	40.0913	0.120240
400.0225	40.2145	0.126000
401.0225	40.3434	0.131400
402.0225	40.4771	0.136080
403.0225	40.6149	0.139680
404.0225	40.7559	0.142200
405.0225	40.8989	0.143640
406.0225	41.0428	0.144000
407.0225	41.1864	0.142920
408.0225	41.3283	0.140400
409.0225	41.4670	0.136440
410.0225	41.6008	0.130680
411.0225	41.7278	0.123120
412.0225	41.8462	0.113400
413.0225	41.9539	0.101520
414.0225	42.0485	0.087480
415.0225	42.1276	0.070560
416.0225	42.1887	0.051120
417.0225	42.2288	0.028800
418.0225	42.2451	0.003240
419.0225	42.2343	-0.025560
420.0225	42.1930	-0.057600
421.0225	42.1176	-0.093600
422.0225	42.0041	-0.133920
423.0225	41.8486	-0.177840
424.0225	41.6467	-0.226800
425.0225	41.3938	-0.280080
426.0225	41.0850	-0.338400
427.0225	40.7151	-0.402120
428.0225	40.2789	-0.471240

429.0225	39.7705	-0.546480
430.0225	39.1839	-0.627840
431.0225	38.5128	-0.715680
432.0225	37.7505	-0.810360
433.0225	36.8900	-0.911880
434.0225	35.9239	-1.021320
435.0225	34.8445	-1.138680
436.0225	33.6438	-1.264320
437.0225	32.3131	-1.398600
438.0225	30.8436	-1.541880
439.0225	29.2259	-1.984880
440.0225	27.4504	-2.547960
441.0225	25.5069	-2.974120
442.0225	14.0313	-5.545440
443.0225	06.0023	-6.104360

Table A-3. Output of Analysis of Sample Data Set 2.

Time (sec)	Distance (m)	Velocity (Kmph)	Acceleration (Kmph/s)
66.4300	452.9400	31.0320	1.298880
72.2700	499.8300	34.4520	1.102680
76.1700	547.7800	41.9040	0.592920
79.9100	590.8900	43.7760	0.423000
82.8300	657.8400	45.1440	0.271080
85.7800	724.5100	46.3320	-0.007920
88.9700	792.4800	46.1880	-0.067680
92.2700	857.6400	45.0000	-0.173880
95.4600	927.6100	43.8480	-0.195480
99.3100	993.9600	42.6960	-0.187200
102.3700	1048.9000	42.3360	-0.178920
106.3700	1101.8300	41.0040	-0.119160
108.7100	1163.9600	40.3200	0.019080
110.8500	1210.1600	40.4280	0.054360
112.9000	1260.1700	42.6240	0.196920
115.1900	1310.1400	42.9840	0.205560
117.5400	1367.8200	45.0720	0.213840
119.7400	1428.9600	46.3320	0.176400
121.9300	1475.7900	47.4120	0.038520
124.0900	1526.2900	47.4480	-0.002160
126.2500	1581.4800	47.3220	-0.042120
130.8400	1627.9000	47.2640	-0.306360
132.6900	1660.0000	47.2340	-2.253240
134.2900	1691.0000	45.2680	-3.335760
136.7900	1734.9000	42.2320	-3.705120
138.4900	1701.8400	40.2320	-5.761800
139.3400	1735.7900	31.1960	-6.513480
141.5510	1744.5990	24.8920	-7.128720
143.0670	1761.7800	16.0478	-7.236000
145.0680	1777.7900	0.0000	0.000000
153.4500	1789.8000	0.0000	0.000000
157.1400	1713.9800	12.0990	1.392400
160.1000	1724.8900	27.6780	3.444200
164.9600	1745.0000	36.6240	3.951800
168.4000	1765.4400	42.3040	4.554440
171.4300	1819.8800	42.7800	4.540160
174.4300	1977.0600	36.9720	-0.661320

182.2700	2047.0600	39.7800	0.619920
193.1100	2217.0600	42.8400	0.188640
196.7200	2257.0200	41.6160	-0.473040
200.0800	2299.4900	39.7800	-0.602280
203.3500	2336.5400	37.8720	-0.527040
205.3700	2358.6600	36.9360	-0.350280
206.8800	2366.0000	36.5760	-0.140760
207.8800	2371.0000	36.5040	0.038520
208.0100	2378.9000	36.5400	0.064080
208.3500	2391.2900	36.5400	0.134280
214.5800	2458.6400	43.0920	2.228760
218.2800	2498.6400	54.9720	4.326120
231.7800	2639.9900	43.3800	-0.246960
233.8800	2667.0000	42.7680	-0.333720
236.9600	2698.5800	41.5800	-0.431640
237.7200	2706.7500	41.2560	-0.450720
238.6000	2716.0000	40.8600	-0.470160
239.2600	2722.8400	40.5360	-0.483120
249.1400	2825.4900	35.3520	-0.524880
253.3800	2869.5400	33.2280	-0.469800
255.7700	2890.7400	32.1480	-0.424080
256.6700	2899.8900	31.7880	-0.404280
257.5700	2906.7100	31.4280	-0.383400
258.1300	2911.6700	31.2120	-0.369720
259.2200	2921.3400	30.8160	-0.342360
259.8500	2926.9300	30.6000	-0.325800
260.3200	2931.1000	30.4560	-0.312840
261.9000	2945.1100	29.9880	-0.268560
263.4000	2958.0400	29.6280	-0.223920
263.9800	2962.9900	29.5200	-0.206280
264.5700	2968.2000	29.3760	-0.187920
269.5700	3020.1500	28.8360	-0.025560
272.8400	3047.0600	28.9440	0.083880
273.6400	3053.4600	29.0160	0.110880
280.2500	3093.5900	30.4560	0.321480
282.4400	3106.8800	31.2480	0.384480
283.2600	3111.8600	31.5720	0.406800
284.7300	3120.7800	32.1840	0.444600
285.5500	3126.5000	32.5800	0.464400
285.8700	3128.8200	32.7240	0.471960
290.2700	3175.6000	34.9920	0.558720
298.4400	3275.5200	39.8880	0.616680
299.3000	3285.7100	40.4280	0.613080

300.8400	3302.9700	41.3640	0.601920
301.7700	3313.0400	41.9040	0.591480
302.1200	3316.9000	42.1200	0.587160
311.7100	3436.8700	46.6560	0.302040
313.0800	3453.1600	47.0160	0.232920
313.5000	3458.0700	47.1240	0.210240
314.1400	3465.3700	47.2320	0.174240
315.5900	3483.0000	47.4120	0.086040
318.7600	3519.9600	47.3400	-0.139320
320.5500	3540.5500	46.9800	-0.287640
321.7900	3554.6700	46.5480	-0.399240
322.8400	3566.5400	46.0800	-0.500040
329.5700	3641.5800	40.2120	-1.285920
332.0000	3698.0000	36.6480	-1.632600
334.0000	3711.0000	33.0840	-1.945080
336.7400	3721.5200	27.1440	-2.413800
337.2600	3727.0300	25.8480	-2.508480
338.6000	3741.2200	22.3200	-2.760120
339.0700	3746.2000	41.6880	-0.331920
340.5600	3761.9800	41.2200	-0.292680
340.6200	3762.6100	41.1840	-0.290880
341.9900	3777.1200	40.8240	-0.257040
342.4600	3782.1000	40.7160	-0.245520
343.1000	3788.9400	40.5360	-0.230760
349.4500	3862.7900	39.5280	-0.102240
352.4600	3897.8000	39.2760	-0.053640
354.7300	3921.3800	39.2040	-0.022320
366.2800	4041.3900	39.6360	0.079200
375.9100	4141.4400	40.5360	0.099360
383.6800	4240.5100	41.2560	0.081720
385.5700	4265.5300	41.4000	0.073800
385.9500	4270.3900	41.4000	0.072360
387.3000	4285.6400	41.5080	0.065520
387.7400	4290.4000	41.5440	0.063360
387.8700	4291.7900	41.5440	0.062640
388.9400	4303.2200	41.6160	0.056880
390.0400	4314.9700	41.6520	0.050400
391.1500	4326.8200	41.7240	0.043560
391.6100	4331.7400	41.7240	0.040680
392.7000	4343.3800	41.7960	0.033480
394.1300	4358.6500	41.8320	0.023760
395.0700	4368.6900	41.8320	0.016920
396.7800	4386.6100	41.8680	0.004320

403.0300	4465.3600	41.7240	-0.046080
404.9700	4490.6400	41.6160	-0.063000
408.2900	4527.1600	41.3640	-0.092160
417.3900	4627.2600	40.1760	-0.171720
418.2700	4636.9400	39.9960	-0.179280
420.0200	4656.0100	39.6720	-0.193680
420.4800	4661.0100	39.6000	-0.197640
421.3600	4670.5400	39.4200	-0.204840
422.2900	4680.5500	39.2400	-0.212040
423.8400	4697.5900	38.8800	-0.223920
427.2900	4735.5000	38.0520	-0.248760
428.3900	4758.9000	37.8000	-0.256320
431.5700	4788.0000	36.9360	-0.275760
436.2600	4801.9000	35.6040	-0.298440
439.5300	4867.5500	34.5960	-0.309960
440.5000	4889.0900	34.3080	-0.672480
441.6400	4909.1000	28.0042	-1.385280
445.2400	4932.8900	20.0980	-3.456000
446.9800	4954.1234	13.7456	-3.699360
448.5400	4969.6708	07.0921	-3.844080
449.7400	4974.0789	0.00000	0.000000

Table A-4. Detailed Analysis of Data Set 2 at Time Interval of One Second

Time (sec)	Velocity (Kmph)	Acceleration (Kmph/s)
Analysis of data just before the first halt		
135.0670	47.3008	-3.371
136.0670	47.3684	-3.543
137.0670	47.4168	-3.975
138.0670	47.4449	-5.646
139.0670	47.4511	-6.351
140.0670	47.4342	-6.531
141.0670	26.8072	-6.729
142.0670	20.3255	-7.179
143.0670	16.0748	-7.236
144.0670	07.7428	-7.128

Analysis of data just before the Last Halt.		
436.000	35.0190	-0.286
437.000	35.6434	-0.296
438.000	35.7010	-0.301
439.000	35.8342	-0.304
440.00	34.3194	-0.483
441.000	30.1963	-0.995
442.000	28.0000	-1.386
443.000	27.1432	-1.427
444.000	24.7534	-2.051
445.000	22.0980	-2.752
446.000	17.0042	-3.560
447.000	13.7456	-3.700
448.000	09.4234	-3.608
449.000	04.0927	-3.922

Table A-5 Result of Analysis of Data Set 3

Time (sec)	Distance (m)	Velocity (Kmph)	Acceleration (Kmph/s)
921.2200	641.0000	59.9040	-0.202680
921.7200	650.0000	59.7960	-0.204120
924.2200	691.0000	59.2920	-0.189720
924.7200	699.0000	59.2200	-0.182880
925.2200	708.0000	59.1120	-0.174240
925.7200	716.0000	59.0400	-0.164160
926.2200	724.0000	58.9680	-0.152640
926.7200	733.0000	58.8960	-0.139680
927.2200	741.0000	58.8240	-0.125280
927.7200	749.0000	58.7520	-0.109440
928.2200	758.0000	58.7160	-0.091800
928.7200	766.0000	58.6800	-0.072720
929.2200	774.0000	58.6440	-0.052200
929.7200	782.0000	58.6080	-0.030240
930.2200	791.0000	58.6080	-0.006480
930.7200	799.0000	58.6080	0.018720
930.7800	800.0000	58.6080	0.021960
931.2800	808.0000	58.6440	0.048960
931.7800	817.0000	58.6440	0.077400
932.2800	825.0000	58.7160	0.107640
932.7800	833.0000	58.7520	0.139680
933.2800	841.0000	58.8600	0.173160
933.7800	858.0000	58.9320	0.208080
934.2800	866.0000	59.0400	0.245160
934.7800	875.0000	59.1840	0.283680
935.2800	891.0000	59.3280	0.323640
936.2800	899.0000	59.7240	0.409320
937.2800	908.0000	60.1560	0.501480
939.2800	924.0000	61.3800	0.707400
939.7800	946.0000	61.7400	0.763200
940.2800	965.0000	62.1360	0.820800
941.7800	979.0000	63.5040	1.005120
942.7600	995.0000	64.5480	1.134360
944.8900	1010.0000	67.2840	1.440360
947.8500	1120.0000	72.2520	1.924560
950.9000	1121.0000	0.0000	0.000000

Table A-6 Output of Detailed Analysis of Data Set 3

Time (sec)	Velocity (Kmph)	Acceleration (Kmph/s)
933.9825	58.9870	0.222840
934.9825	59.2477	0.299520
935.9825	59.5885	0.383040
936.9825	60.0161	0.473400
937.9825	60.5374	0.570600
938.9825	61.1595	0.675000
939.9825	61.8895	0.786240
940.9825	62.7346	0.905040
941.9825	63.7020	1.031040
942.9825	64.7992	1.164600
943.9825	66.0338	1.305720
944.9825	67.4133	1.454400
945.9825	68.9455	1.611000
946.9825	70.6381	1.775520
947.9825	72.3992	1.925280
948.9825	72.5368	1.947960
949.9825	72.7590	1.987560

Photo 1. Photo of the Start up Panel.

Photo 2. Photo showing the Plot of Time Vs Velocity.

Photo 3. Photo showing Plot of Time Vs Velocity of Another Data Set

Photo 4. Photo showing a part of the plot after Zooming In.

Photo. 5. Photo showing the XY Graph Popup- plot of Time Vs Acceleration.

REFERENCES

1. Neil, Bennet, Homepage of Penny & Giles Aerospace Ltd., Internet, March 1996.
2. Rajpal, Ashok, " A PC Based digital Flight Data Recorder", M.Tech Thesis, Aeronautical Engineering Department, IIT Kanpur, December 1990.
3. Mohit, K. and Pradeep, B., "Design and Development of a Flight Data Acquisition System", B. Tech Report, Aeronautical Engineering Department, IIT Kanpur, April 1983.
4. Homepage of National Instruments Corporation, July 1996.
5. LabWindows/CVI User Interface Reference Manual, National Instruments Corporation, Austin TX, 1994.
6. LabWindows/CVI Advanced Analysis Reference Manual, National Instruments Corporation, Austin TX, 1994.
7. LabWindows/CVI Standard Libraries Reference Manual, National Instruments Corporation, Austin TX, 1994.
8. LabWindows/CVI Programmers Reference Manual, National Instruments Corporation, Austin TX, 1994.
9. Reddy, I. Ravinder., "Simulation of Heterogeneous Traffic on Narrow Roads", M.Tech Thesis, Department of Civil Engineering, IIT Kanpur, February 1996.
10. Press ,William H., "Numerical Recipes in C", Cambridge University Press, 1990.